# BRIDGE: A Program and User's Manual for Generating Summary Catch Log Reports

Peter A. Slootweg and Donald J. Noakes

Department of Fisheries and Oceans
Fisheries Research Branch
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

January 1988

**Canadian Technical Report of Fisheries and Aquatic Sciences No. 1596**

Fisheries and Oceans Pêches et Océans

Canada

88-04865/

## Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

## Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annual des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Technical Report of
Fisheries and Aquatic Sciences No. 1596

January 1988

# BRIDGE: A Program and User's Manual for Generating Summary Catch Log Reports

by

Peter A. Slootweg and Donald J. Noakes

Department of Fisheries and Oceans
Fisheries Research Branch
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

(C)Minister of Supply and Services Canada 1988

Cat. No. Fs 97-6/1596E ISSN 0706-6457

Slootweg, P. A. and D. J. Noakes. 1988. Bridge: A program and user's manual for generating summary catch log reports. Can. Tech. Rep. Fish. Aquat. Sci. 1596: 71 p.

## ABSTRACT

Slootweg, P. A. and D. J. Noakes. 1988. Bridge: A program and user's manual for generating summary catch log reports. Can. Tech. Rep. Fish. Aquat. Sci. 1596: 71 p.

The most basic and generally the first piece of information reported from a fisheries research survey or observer program is a summary of catch. The program presented in this report generates summaries of catch and catch per unit effort (CPUE) in terms of weight and/or pieces by species/species group and sampling unit. The program is written in standard FORTRAN 77 (ANSI X3.9-1978) and has been successfully implemented on a VAX 11/780 using VAX FORTRAN 77 and on an IBM compatible microcomputer employing both Microsoft FORTRAN 77 V4.0 and Lahey Personal FORTRAN 77 V1.0. In addition to being portable, the source code is well documented and should allow users to easily customize the program to suit their particular needs. Copies of the program can be obtained by contacting the second author.

## RÉSUMÉ

Slootweg, P. A. and D. J. Noakes. 1988. Bridge: A program and user's manual for generating summary catch log reports. Can. Tech. Rep. Fish. Aquat. Sci. 1596: 71 p.

Les renseignements les plus fondamentaux et généralement les premiers à être communiqués après la fin d'un relevé de recherche ou d'un programme de vérification par observateurs des pêches se présentent sous la forme d'un résumé des prises. Le programme décrit dans le présent rapport permet d'obtenir des résumés des prises et des prises par unité d'effort (PUE) se présentant sous la forme de la masse ou du nombre de poissons par espèces, groupes d'espèces ou unités d'échantillonnage. Le programme est rédigé en FORTRAN 77 standard (ANSI X3.9 - 1978) et a été utilisé avec succès sur VAX 11/17, à l'aide d'un VAX FORTRAN 77, et sur micro-ordinateur compatible au IBM, à l'aide du Microsoft FORTRAN 77 V4.0 et du Lahey Personal FORTRAN 77 V1.0. En plus d'être portable, le code de sources est bien étayé et devrait permettre aux utilisateurs d'adapter facilement le programme à leurs besoins particuliers. Des exemplaires du programme peuvent être obtenus en s'adressant aux second auteur.

## INTRODUCTION

The most common method of obtaining fishery independent estimates of stock status is to conduct research surveys using standardized gear and procedures. Catch is generally recorded in terms of weight and/or number of pieces by species for each discrete sampling unit (e.g. set, tow, etc.). A "trip report" consists of, among other things, a summary of catch by species for each sampling unit. Information such as date, time, location, and depth is taken directly from the vessel's "bridge log" and reported along with the observed catches and CPUE's for each sampling unit.

The computer program presented in this report has two attractive characteristics. First, the program is written in standard ANSI (X3.9-1978) FORTRAN 77. This may appear to be a trivial point but it will certainly be appreciated by users that have attempted to convert programs from one computer system to another. The only detail that should concern the user is how the "INCLUDE" files are handled by the particular FORTRAN compiler employed on their system (users should refer to their language reference manual to determine if and how INCLUDE statements are employed by their FORTRAN compiler). If the INCLUDE statement is not supported by their compiler, then the code in the INCLUDE files can be incorporated into the appropriate subroutines.

The second attractive characteristic is, in fact, related to the first. Because the code is portable, it can be easily implemented on an IBM compatible microcomputer. The wide spread availability of these machines means that trip reports could be completed by the time the ship reaches port.

The remainder of this report deals with the inputs, outputs, and operation of the program. The program listing presented in this report has been adapted to handle unique features employed in surveys conducted by researchers studying groundfish species in the Pacific Region. Two other versions of the program have been developed to address specific needs of researchers assessing shrimp and flying squid stocks in B.C. coastal waters. A complete listing of the program, input data, and output employed to demonstrate the function of the program are included in this report.

## PROGRAM INPUT

Seven files are used to enter the input data into the program. These files are required by the program and normally reside in the same directory. Six files (Appendix tables A-F) are used to initialize variables and select options. The seventh file (Appendix table G) contains the survey data to be summarized. Users may elect to copy the

default files (Appendix tables A-F) supplied by the system to their own directory or create custom files using the defaults as guides. The survey data file (Appendix table G) is supplied by the user.

Some of the information contained in these files could have been incorporated directly into the code. This would, however, restrict the user's ability to easily modify the output. Employing files to input this information means the program does not have to be altered and recompiled each time a minor change is required.

## FILE 1: TRIP DATA

This file contains the bridge log and catch information for the survey. Two separate forms are employed to record these data. The first sheet, the bridge log (Fig. 1), is used to record such information as the name of the vessel, the type of gear employed, and the date, time, and location where fishing occurred. The second sheet, the catch log (Fig. 2), is used to record the catch for each species in terms of weight and/or number of pieces. Separate bridge and catch logs are completed for each sampling unit in the survey. Each sample will have exactly one bridge log (generally identified as card code 01 in columns 79 and 80) and a separate catch log for each species caught in the sample (generally identified as card code 02 (shellfish) or 03 (groundfish) in columns 79 and 80). The program reads each record, determines whether it is a bridge or catch log, and proceeds accordingly. The program is setup to read the data in the formats shown in Figures 1 and 2 but could be easily modified by simply changing the appropriate format statements in the subroutine DATAINIT (Appendix: Program Listing).

## FILE 2: FISH CODE

It is common practice to employ alphanumeric codes in place of common or scientific species names. Researchers at the Pacific Biological Station have adopted a three character code based on the identification key presented by Hart (1973). For instance, the species code for Pacific ocean perch (Sebastes alutus) is 396 (Fig. 2) which corresponds to the page in Hart (1973) where this species is described. Species not appearing in Hart (1973) have been assigned unique codes by researchers studying that particular species. These codes appear on the catch log sheet (Fig. 2) and are employed instead of the species name.

The "fish code" file (Appendix table A) contains three pieces of information employed by the program: the species or group number, the three character alphanumeric species code, and the common or scientific name of the species. The species or group number is arbitrary and only affects the appearance of the output. The number is assigned to a set of species that are related in some sense. For instance, in Appendix table A, all invertebrates have been assigned group number 4. All output

related to these species will appear in the same physical location enabling the investigator to clarify or demonstrate observations related to the survey results. The program currently allows for a maximum of ten (10) groups to be identified although in practice only five or six are generally employed.

The program employs the alphanumeric codes to sort and summarize the catch by species. Cross-referencing in the "fish code" file allows for the conversion of the codes to the common or scientific names thus ensuring the output is more readable.

FILE 3: ESTIMATED AVERAGE WEIGHT (EST. AVE.)

A sample "estimated average weight" file is shown in Appendix table B. This file contains estimated average weights (kgs) for the species found in the fish code file. These averages generally reflect long term observed average weights assembled from previous surveys. In some instances, investigators may elect to sub-sample the catches from the current survey to estimate average weights in order to reflect any spatial or seasonal changes.

In some surveys, landings of each species are recorded only in terms of total landed weight. The average weights contained in the "estimated average weight" file are then employed to estimate the number of pieces landed for each species. Quantities that have been estimated using this procedure are clearly marked with an "*" in the output. The same method of calculation applies to surveys where only the number of pieces caught have been recorded.

FILE 4: GROUP NAME

An example "group name" file is presented in Appendix table C. The "group name" file provides a short (16 characters or less) description of each of the 10 groups identified in the fish code file. Group numbers not employed in the "fish code" file should be assigned the same description, usually "OTHERS".

FILE 5: LINE TITLE

A sample "line title" file is shown in Appendix table D. The "line title" file contains some of the headings employed in the output. This information could have been included directly in the program source code but a separate file was used to provide additional flexibility to the user. For instance, if the survey recorded fishing location in terms of Loran C instead of latitude and longitude, the appropriate heading in

the "line title" file could be changed instead of altering and recompiling the program.

FILE 6: REMARKS

A sample "remarks" file is presented in Appendix table E. The "remarks" file contains a list of possible remarks the investigator may wish to use to describe certain aspects of the sample. A numeric code is employed to identify if the sample is usable or if some problem (i.e. gear damaged) existed with the particular sample.

FILE 7: INIT.DAT

The "init.dat" file (Appendix table F) contains a list of the input and output files to be employed by the program as well as the initial values assigned to various parameters. This file is read by the program and the information contained in this file is used to initialize filenames and variables. If certain options are changed by the user, the program provides the option of writing the new selections back to the "init.dat" file so they may be used as defaults in the future.

Each piece of information in the "init.dat" file is entered on a separate line. The first seven lines contain the names of the input and output files in the order: 1) fish code file; 2) estimated average weight file; 3) trip data file; 4) group name file; 5) line title file; 6) remarks file; and 7) the output file. The next three (3) lines (8, 9, and 10) in the "init.dat" file provide the page header to be used in the report.

The remaining three lines (11, 12, and 13) are used to control the appearance of the output. The program allows the user to specify the maximum number of species to appear in the output for each of the 10 possible groups (default set to 100 for each group). The user may, for instance, only be interested in obtaining information on the top five species, either in terms of weight or number of pieces, for each group and lumping all other landings in that group under "OTHERS". This can be accomplished by identifying the maximum number of species to appear in each of the 10 groups on line 11 in format 10I3 and entering a "P" or "W" on line 12. For example, the "init.dat" file presented in Appendix table F instructs the program to output information on the top 15, 10, and 5 species in groups 1, 2, and 3, respectively, (i.e. line 11 is 15 10 5 0100100100100100100) in terms of weight (i.e. line 12 is "W"). Landings for all invertebrate species (group 4) are lumped together as "OTHERS" (i.e. setting for group 4 on line 11 is 0). Since only 4 groups have been identified, the settings for groups 5 through 10 are left at the default of 100.

The program can generate up to six different summary reports. The catch information for each sampling unit can be presented in terms of weight (Report 1), weight/km (Report 2), weight/km*hr (Report 3), pieces (Report 4), pieces/km (Report 5), and pieces/km*hr (Report 6). The last line (line 13) of the "init.dat" file instructs the program which reports are to be generated. For instance, the entry "100000" instructs the program to generate only the first report.

## PROGRAM OPERATION

Two version of the program are available. The "generic" version is completely portable and will run on any computer system that supports the full implementation of FORTRAN 77. A modified program has also been developed to run specifically on VAX computers. The VAX version uses screen management utilities to generate windows and highlight the current activity and associated options (Kuhn 1987). The available options and output for both programs are identical. Both versions of the program are menu driven.

### GENERIC VERSION

The basic menus employed in the generic program are presented in Table 1. Users branch to one of the four (4) sub-menus by selecting the appropriate letter (uppercase only). Users type "F" to change the input or output file names, "G" to alter the number of species to appear in the output for each species grouping and determine whether the selection is on the basis of weight or number of pieces, "P" to adjust the page header, and "R" to select the reports to be generated. Entering "W" writes the selected options back to the "init.dat" file, "S" starts the generation of the desired reports, and "Q" drops the user out of the program. Users are always returned to the main menu after exiting one of the sub-menus.

### VAX VERSION

The screens employed in the VAX version of the program are presented in Table 2. The VAX version employs screen management utilities to generate windows to highlight the current activity and available options (Kuhn 1987). The windows replace the sub-menus used in the generic version of the program. In the main menu, users may select the desired activity by typing the first letter of the unit (uppercase, as in the generic version of the program) or by using the cursor keys to move to the appropriate activity and pressing return. Movement and

control within a sub-activity are invoked by the up and down cursor keys and return. The back slash character, "\", returns the user to the main menu from any of the sub-activities and also exits the program. Selecting the "PF2" key invokes on screen help.

## PROGRAM OUTPUT

The data contained in Appendix tables A through G were used to generate a catch report for a Groundfish Assemblage Survey conducted during January and February 1986. The "trip data" file is presented in Appendix table G and the catch report is presented in Table 3. Only the first 12 samples from this survey are employed to demonstrate the operation of the program.

The first two sections of the catch report (Tables 3a and 3b) provide a detailed summary of the catch log information recorded during the survey. Since the number of pieces caught for each species was not recorded by the survey crew, the pieces and calculated average weights appear as columns of zeros on the first page of the report (Table 3a). Estimated average weights were set equal to one (1) by default if no estimate of average weight was provided in the "estimated average weight" file (Appendix table B). The second page of the report (Table 3b) provides an estimate of the proportion of the catch by species both in terms of landed weight and number of pieces. Since the number of pieces were not recorded and the estimated number of pieces are unreliable, the percentages by pieces are not particularly interesting in this instance. The species are sorted in descending order within groups in these two tables.

The third page of the report (Table 3c) supplies information on the format of the remaining tables. The number of species that qualified to print (according to the user's instructions) are listed by group. In this example, the user has requested that catch information be printed for the first 15, 10, and 5 species in groups 1, 2, and 3 respectively. Landings for all other groups (including group 4, invertebrates) are listed as others.

The remainder of the report (Table 3d) summarizes the catch by species for each sampling unit. This summary appears on two pages with each page covering six discrete sampling units. The first page presents information on the date, time, location, depth, and type of gear employed in the survey. The second page summarizes the catch by species and prints the results by group. The program displays weight and pieces information (actual or estimated) for the target species (the predominant species in group 1). For instance, in this example, the actual weight and estimated number of pieces of English sole are printed for each sampling unit (Table 3d). Only one (Report 1 - Weights) of the six possible reports has been generated in this example.

## SUMMARY REMARKS

The program presented in this report is an effective means of generating catch summaries of survey or observer data. The program is written in standard FORTRAN 77 and can be easily implemented on a variety of computer systems. The program is menu driven and can be easily modified to suit the particular needs of users. The program is available upon request.

## ACKNOWLEDGMENTS

Special thanks to Brian Kuhn for supplying the screen management subroutines employed in the VAX version of the program and to Jeff Fargo for allowing us to use the assemblage survey data as an example. Dwight Heritage and Bill Shaw reviewed a draft of this report and their comments and suggestions are appreciated.

## REFERENCES

Hart, J. L. 1973. Pacific Fishes of Canada. Fish. Res. Board Can. Bull. 180: 740 p.

Kuhn, B. R. 1987. Special screen and miscellaneous VAX programmer utilities reference manual. Pacific Biological Station, Nanaimo, British Columbia. (Unpubl. MS.)

Table 1. Menus for Generic Bridge Log Program.

```
                    MAIN MENU

<F>ILE OPTIONS
<G>ROUPING OPTIONS
<P>AGE HEADER
<R>EPORTS TO BE GENERATED
<S>TART
<W>RITE OPTIONS TO DISK
<Q>UIT
PLEASE ENTER YOUR CHOICE ( F,G,P,R,S,W OR Q)


                  FILE MENU

1.FISH CODE FILE
  SPEC2.DAT
2.ESTIMATED AVERAGE FILE
  FISHAVE.DAT
3.TRIP DATA FILE
  FLATFISH.DAT
4.GROUP NAME FILE
  GROUP.DAT
5.PAGE 1 LINE TITLE FILE
  LINETITLE.DAT
6.CATCH REMARKS FILE
  REMARKS.DAT
7.OUTPUT FILE
  TEST.DAT
 ENTER YOUR CHOICE (DEFAULT = 0 TO QUIT)


                  GROUP MENU

GROUP      MAX NO./GRP
  1            15
  2            10
  3             5
  4             0
  5           100
  6           100
  7           100
  8           100
  9           100
 10           100
 11 <W>EIGHTS OR <P>IECES = W
ENTER n TO CHANGE MAX/GROUP , 11 TO CHANGE <W>EIGHTS OR <P>IECES , 0 TO
QUIT
```

Table 1 (cont'd)

```
                    PAGE HEADER MENU

 1.GROUNDFISH 1986 ASSEMBLAGE SURVEY
 2. R/V G.B. REED
 3. J. FARGO
ENTER YOUR CHOICE (DEFAULT = 0 TO QUIT)


                    REPORT MENU

ON  1. WEIGHTS ( KG)
OFF 2. WEIGHTS ( KG)/( KM)
OFF 3. WEIGHTS ( KG)/( KM*HRS)
OFF 4. PIECES
OFF 5. PIECES / KM
OFF 6. PIECES /(KM*HRS)
ENTER n TO TOGGLE ON/OFF, 0 TO QUIT
```

Table 2. Menus for the VAX Version of the Program.

```
                              MAIN MENU

30-Sep-1987 at  8:00am                BRIDGE LOG          User: Noakes
Wednesday                               PROGRAM

FILE  GROUPING  PAGE_HEADER  REPORTS  START  WRITE

                      DRC0:[NOAKES.PUBLIC]
        PF2 = HELP, ARROWS = Move Cursor, \ = EXIT


                         FILE OPTION MENU

30-Sep-1987 at  8:00am                BRIDGE LOG          User: Noakes
Wednesday                               PROGRAM

SET FILE OPTIONS
FILE  GROUPING  PAGE_HEADER  REPORTS  START  WRITE
 FISH CODE         SPEC2.DAT
 EST. AVE          FISHAVE.DAT
 TRIP DATA         FLATFISH.DAT
 GROUP NAME        GROUP.DAT
 P.1 LINE TITLE    LINETITLE.DAT
 CATCH REMARKS     REMARKS.DAT
 OUTPUT            TEST.DAT

                      DRC0:[NOAKES.PUBLIC]
        PF2 = HELP, ARROWS = Move Cursor, \ = EXIT
```

Table 2 (cont'd)

```
                        GROUP OPTION MENU

30-Sep-1987 at  8:00am          BRIDGE LOG          User: Noakes
Wednesday                        PROGRAM

SET GROUPING OPTIONS
FILE  GROUPING  PAGE_HEADER  REPORTS  START  WRITE
        1 MAX NUM/GRP  15
        2 MAX NUM/GRP  10
        3 MAX NUM/GRP   5
        4 MAX NUM/GRP   0
        5 MAX NUM/GRP 100
        6 MAX NUM/GRP 100
        7 MAX NUM/GRP 100
        8 MAX NUM/GRP 100
        9 MAX NUM/GRP 100
        0 MAX NUM/GRP 100
        WTS. OR PCES = P

                DRCO:[NOAKES.PUBLIC]
          PF2 = HELP, ARROWS = Move Cursor, \ = EXIT
```

```
                      PAGE_HEADER OPTIONS

30-Sep-1987 at  8:00am          BRIDGE LOG          User: Noakes
Wednesday                        PROGRAM

SET PAGE HEADER FOR REPORT
FILE  GROUPING  PAGE_HEADER  REPORTS  START  WRITE

1 GROUNDFISH 1986 ASSEMBLAGE SURVEY
2 R/V G.B. REED
3 J. FARGO

                DRCO:[NOAKES.PUBLIC]
          PF2 = HELP, ARROWS = Move Cursor, \ = EXIT
```

Table 2 (cont'd)

```
                        REPORT OPTIONS

30-Sep-1987 at  8:00am                BRIDGE LOG        User: Noakes
Wednesday                               PROGRAM

SET REPORTS TO BE GENERATED
FILE  GROUPING  PAGE_HEADER  REPORTS  START  WRITE
                              1.  ON -  WEIGHTS ( KG)
                              2. OFF -  WEIGHTS ( KG)/( KM)
                              3. OFF -  WEIGHTS ( KG)/( KM*HRS)
                              4. OFF -  PIECES
                              5. OFF -  PIECES / KM
                              6. OFF -  PIECES / KM*HRS

                  DRCO:[NOAKES.PUBLIC]
       PF2 = HELP, ARROWS = Move Cursor, \ = EXIT
```

Table 3a. Sample Catch Report as produced by program BRIDGE. Summary of catch by weight and pieces for the entire survey listed in descending order (by weight) by group.

TOTAL NUMBER OF HAULS FOR CRUISE = 12
TOTAL CATCH IN WEIGHTS AND PIECES, ESTIMATED AVERAGE WEIGHT.

| GRP | CODE | SPECIES | KG | PIECES | ESTIMATED AVE WT. | CALCULATED AVE WT. |
|---|---|---|---|---|---|---|
| 1 | 628 | ENGLISH SOLE | 445. | 0 | 1.00 | .00 |
| 1 | 602 | TURBOT | 243. | 0 | 1.00 | .00 |
| 1 | 222 | PACIFIC COD | 199. | 0 | 1.00 | .00 |
| 1 | 596 | DAB (PACIFIC) | 170. | 0 | 1.00 | .00 |
| 1 | 467 | LINGCOD | 68. | 0 | 3.00 | .00 |
| 1 | 614 | HALIBUT | 67. | 0 | 10.00 | .00 |
| 1 | 607 | PETRALE SOLE | 50. | 0 | 1.00 | .00 |
| 1 | 610 | REX SOLE | 48. | 0 | 1.00 | .00 |
| 1 | 435 | S. PAUCISPINIS | 19. | 0 | 1.00 | .00 |
| 1 | 621 | ROCK SOLE | 10. | 0 | 1.00 | .00 |
| 1 | 442 | S. RUBERRIMUS | 5. | 0 | 1.00 | .00 |
| 1 | 437 | S. PINNIGER | 2. | 0 | 1.00 | .00 |
| 1 | 631 | STARRY FLOUNDER | 2. | 0 | 1.00 | .00 |
| 1 | 612 | FLATHEAD SOLE | 1. | 0 | 1.00 | .00 |
| 1 | 228 | WALLEYE POLLOCK | 1. | 0 | 1.00 | .00 |
| 1 | 619 | BUTTER SOLE | 0. | 0 | 1.00 | .00 |
| 1 | 636 | SAND SOLE | 0. | 0 | 1.00 | .00 |
| 1 | 635 | CURLFIN SOLE | 0. | 0 | 1.00 | .00 |
| 1 | 626 | DOVER SOLE | 0. | 0 | 1.00 | .00 |
| 1 | 625 | SLENDER SOLE | 0. | 0 | 1.00 | .00 |
| 1 | 446 | S. VARIEGATUS | 0. | 0 | 1.00 | .00 |
| 1 | 598 | DAB (SPECKLED) | 0. | 0 | 1.00 | .00 |
| 1 | 418 | S. FLAVIDUS | 0. | 0 | 1.00 | .00 |
| 2 | 066 | RATFISH | 668. | 0 | .50 | .00 |
| 2 | 056 | BIG SKATE | 168. | 0 | 1.00 | .00 |
| 2 | 044 | DOGFISH | 138. | 0 | 2.50 | .00 |
| 3 | 304 | CYMATOGASTER | 5. | 0 | 1.00 | .00 |
| 3 | 096 | HERRING | 1. | 0 | .12 | .00 |
| 3 | 550 | STURGEON POACHER | 0. | 0 | 1.00 | .00 |
| 4 | 999 | UNKNOWN INVERTEB | 77. | 0 | 1.00 | .00 |
| 4 | 92A | SQUID | 66. | 0 | 1.00 | .00 |
| 4 | 5QA | BASKET STARS | 4. | 0 | 1.00 | .00 |
| 4 | SQA | EUALUS SPP. | 3. | 0 | 1.00 | .00 |
| 4 | 6AB | SEA URCHINS | 2. | 0 | 1.00 | .00 |
| 4 | 4GA | STARFISH | 2. | 0 | 1.00 | .00 |
| 4 | 3L0 | ANEMONE | 0. | 0 | 1.00 | .00 |
| 4 | VAC | HERMIT CRAB | 0. | 0 | 1.00 | .00 |
| 4 | VAA | ANOMURA | 0. | 0 | 1.00 | .00 |
| 4 | SAB | SHRIMP | 0. | 0 | 1.00 | .00 |

TOTAL KG= 2464. TOTAL PIECES = 0 NUMBER OF SPECIES PRESENT = 39

Table 3b. Sample Catch Report as produced by program BRIDGE. Summary of catch presented in terms of percent by weight and pieces for each species. Listed in descending order (by weight) by group.

TOTAL NUMBER OF HAULS FOR CRUISE = 12
TOTAL CATCH IN WEIGHTS AND PIECES, ESTIMATED AVERAGE WEIGHT.

| GRP | CODE | SPECIES | KG | PERCENT | PIECES | PERCENT | AVERAGE WT/PIECE |
|---|---|---|---|---|---|---|---|
| 1 | 628 | ENGLISH SOLE | 445. | 18.06 | 445* | 15.12 | 1.00 |
| 1 | 602 | TURBOT | 243. | 9.86 | 243* | 8.25 | 1.00 |
| 1 | 222 | PACIFIC COD | 199. | 8.08 | 199* | 6.76 | 1.00 |
| 1 | 596 | DAB (PACIFIC) | 170. | 6.90 | 170* | 5.77 | 1.00 |
| 1 | 467 | LINGCOD | 68. | 2.76 | 21* | .71 | 3.24 |
| 1 | 614 | HALIBUT | 67. | 2.72 | 4* | .14 | 16.75 |
| 1 | 607 | PETRALE SOLE | 50. | 2.03 | 50* | 1.70 | 1.00 |
| 1 | 610 | REX SOLE | 48. | 1.95 | 48* | 1.63 | 1.00 |
| 1 | 435 | S. PAUCISPINIS | 19. | .77 | 19* | .65 | 1.00 |
| 1 | 621 | ROCK SOLE | 10. | .41 | 10* | .34 | 1.00 |
| 1 | 442 | S. RUBERRIMUS | 5. | .20 | 5* | .17 | 1.00 |
| 1 | 437 | S. PINNIGER | 2. | .08 | 2* | .07 | 1.00 |
| 1 | 631 | STARRY FLOUNDER | 2. | .08 | 2* | .07 | 1.00 |
| 1 | 612 | FLATHEAD SOLE | 1. | .04 | 1* | .03 | 1.00 |
| 1 | 228 | WALLEYE POLLOCK | 1. | .04 | 1* | .03 | 1.00 |
| 1 | 619 | BUTTER SOLE | 0. | .00 | 0 | .00 | .00 |
| 1 | 636 | SAND SOLE | 0. | .00 | 0 | .00 | .00 |
| 1 | 635 | CURLFIN SOLE | 0. | .00 | 0 | .00 | .00 |
| 1 | 626 | DOVER SOLE | 0. | .00 | 0 | .00 | .00 |
| 1 | 625 | SLENDER SOLE | 0. | .00 | 0 | .00 | .00 |
| 1 | 446 | S. VARIEGATUS | 0. | .00 | 0 | .00 | .00 |
| 1 | 598 | DAB (SPECKLED) | 0. | .00 | 0 | .00 | .00 |
| 1 | 418 | S. FLAVIDUS | 0. | .00 | 0 | .00 | .00 |
| 2 | 066 | RATFISH | 668. | 27.11 | 1336* | 45.38 | .50 |
| 2 | 056 | BIG SKATE | 168. | 6.82 | 168* | 5.71 | 1.00 |
| 2 | 044 | DOGFISH | 138. | 5.60 | 53* | 1.80 | 2.60 |
| 3 | 304 | CYMATOGASTER | 5. | .20 | 5* | .17 | 1.00 |
| 3 | 096 | HERRING | 1. | .04 | 8* | .27 | .13 |
| 3 | 550 | STURGEON POACHER | 0. | .00 | 0 | .00 | .00 |
| 4 | 999 | UNKNOWN INVERTEB | 77. | 3.13 | 77* | 2.62 | 1.00 |
| 4 | 92A | SQUID | 66. | 2.68 | 66* | 2.24 | 1.00 |
| 4 | 5QA | BASKET STARS | 4. | .16 | 4* | .14 | 1.00 |
| 4 | SQA | EUALUS SPP. | 3. | .12 | 3* | .10 | 1.00 |
| 4 | 6AB | SEA URCHINS | 2. | .08 | 2* | .07 | 1.00 |
| 4 | 4GA | STARFISH | 2. | .08 | 2* | .07 | 1.00 |
| 4 | 3L0 | ANEMONE | 0. | .00 | 0 | .00 | .00 |
| 4 | VAC | HERMIT CRAB | 0. | .00 | 0 | .00 | .00 |
| 4 | VAA | ANOMURA | 0. | .00 | 0 | .00 | .00 |
| 4 | SAB | SHRIMP | 0. | .00 | 0 | .00 | .00 |

TOTAL KG= 2464. TOTAL PIECES = 2944 NUMBER OF SPECIES PRESENT = 39

Table 3c. Sample Catch Report as produced by program BRIDGE. Number and names of species to be printed for each group as specified by the user.

SPECIES TO QUALIFY TO PRINT

| GROUP | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| MAXIMUM SPECIES/GRP | 15 | 10 | 5 | 0 | 100 | 100 | 100 | 100 | 100 | 100 |

| GRP | CODE | NAME |
|---|---|---|
| 1 | 628 | ENGLISH SOLE |
| 1 | 602 | TURBOT |
| 1 | 222 | PACIFIC COD |
| 1 | 596 | DAB (PACIFIC) |
| 1 | 467 | LINGCOD |
| 1 | 614 | HALIBUT |
| 1 | 607 | PETRALE SOLE |
| 1 | 610 | REX SOLE |
| 1 | 435 | S. PAUCISPINIS |
| 1 | 621 | ROCK SOLE |
| 1 | 442 | S. RUBERRIMUS |
| 1 | 437 | S. PINNIGER |
| 1 | 631 | STARRY FLOUNDER |
| 1 | 612 | FLATHEAD SOLE |
| 1 | 228 | WALLEYE POLLOCK |
| 2 | 066 | RATFISH |
| 2 | 056 | BIG SKATE |
| 2 | 044 | DOGFISH |
| 3 | 304 | CYMATOGASTER |
| 3 | 096 | HERRING |
| 3 | 550 | STURGEON POACHER |

Table 3d. Sample Catch Report as produced by program BRIDGE. Summary of catch by sampling unit.

APPENDIX TABLE 1 - WEIGHTS ( KG) PAGE 1.1
GROUNDFISH 1986 ASSEMBLAGE SURVEY
R/V G.B. REED
J. FARGO

| SET NO. | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| DATE | JAN. 29 | JAN. 29 | JAN. 30 | JAN. 30 | JAN. 30 | JAN. 31 |
| AREA | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| TIME START (LST) | 1212 | 1330 | 828 | 1340 | 1518 | 829 |
| DURATION(HR.MIN) | .30 | .30 | .30 | .30 | .30 | .30 |
| START N. LAT. (DEG) | 53 | 53 | 53 | 53 | 53 | 53 |
| (MIN) | 5.7 | 5.2 | 5.1 | 10.9 | 10.2 | 9.7 |
| W. LONG. (DEG) | 130 | 130 | 130 | 130 | 130 | 130 |
| (MIN) | 52.9 | 53.3 | 53.2 | 47.3 | 47.6 | 47.9 |
| DIRECTION (DEG.TRUE) | 309 | 306 | 312 | 337 | 336 | 340 |
| FINISH N. LAT. (DEG) | 53 | 53 | 53 | 53 | 53 | 53 |
| (MIN) | 6.8 | 6.2 | 6.1 | 12.6 | 11.8 | 11.0 |
| W.LONG. (DEG) | 130 | 130 | 130 | 130 | 130 | 130 |
| (MIN) | 55.0 | 55.7 | 54.9 | 48.5 | 48.7 | 48.7 |
| LENGTH OF SET KM. | 1.6 | 1.8 | 1.5 | 1.8 | 1.8 | 1.4 |
| DEPTH START (m-fa) | 88- 48 | 79- 43 | 77- 42 | 130- 71 | 130- 71 | 130- 71 |
| FINISH (m-fa) | 84- 45 | 75- 41 | 77- 42 | 124- 67 | 128- 69 | 126- 68 |
| INTERV. (m-fa) | 80- 40 | 70- 40 | 70- 40 | 120- 60 | 120- 70 | 120- 60 |
| SEE FIGURE NO. | | | | | | |
| START TEMP.(DEG.C) | .0 | .0 | .0 | .0 | .0 | .0 |
| END TEMP.(DEG.C) | .0 | .0 | .0 | .0 | .0 | .0 |
| TYPE OF GEAR | 0 | 0 | 0 | 0 | 0 | 0 |
| NO. OF GROUPS. | 0 | 0 | 0 | 0 | 0 | 0 |
| TOTAL CATCH (KG) | 197 | 172 | 277 | 227 | 91 | 165 |
| REMARKS | USABLE | USABLE | USABLE | USABLE | USABLE | USABLE |

Table 3d (cont'd)

APPENDIX TABLE 1 - WEIGHTS ( KG)
GROUNDFISH 1986 ASSEMBLAGE SURVEY
R/V G.B. REED
J. FARGO

PAGE 1.2

| SET NO. | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| DATE | JAN. 29 | JAN. 29 | JAN. 30 | JAN. 30 | JAN. 30 | JAN. 31 |
| AREA | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| TOTAL CATCH WEIGHT | 199 | 176 | 278 | 250 | 101 | 186 |
| TOTAL CATCH PIECES | 184* | 158* | 278* | 205* | 66* | 175* |
| GROUNDFISH | | | | | | |
| ENGLISH SOLE | 116 | 35 | 138 | 60 | 11 | 34 |
| PIECES | 116* | 35* | 138* | 60* | 11* | 34* |
| TURBOT | TRACE | TRACE | .. | 16 | 2 | 7 |
| PACIFIC COD | 6 | 10 | 12 | 13 | 11 | 28 |
| DAB (PACIFIC) | 45 | 44 | 81 | .. | .. | .. |
| LINGCOD | 18 | 8 | TRACE | 9 | 13 | 2 |
| HALIBUT | 4 | 13 | .. | 5 | 11 | .. |
| PETRALE SOLE | .. | .. | TRACE | 27 | 1 | 9 |
| REX SOLE | TRACE | TRACE | .. | 14 | 5 | 20 |
| S. PAUCISPINIS | .. | .. | .. | .. | .. | 2 |
| ROCK SOLE | 1 | 1 | 3 | .. | .. | .. |
| S. RUBERRIMUS | .. | .. | .. | .. | .. | .. |
| S. PINNIGER | .. | .. | .. | .. | .. | .. |
| STARRY FLOUNDER | .. | .. | .. | .. | .. | .. |
| FLATHEAD SOLE | .. | .. | .. | .. | .. | .. |
| WALLEYE POLLOCK | TRACE | .. | .. | .. | .. | .. |
| OTHERS | .. | TRACE | TRACE | TRACE | .. | TRACE |
| SELACHI | | | | | .. | TRACE |
| RATFISH | 1 | .. | .. | 1 | TRACE | TRACE |
| BIG SKATE | 4 | 57 | 41 | .. | .. | 14 |
| DOGFISH | .. | .. | .. | 57 | 26 | 26 |
| OTHERS | .. | .. | .. | .. | .. | .. |
| FISH | | | | | | |
| CYMATOGASTER | .. | TRACE | 1 | 2 | 1 | 1 |
| HERRING | .. | .. | .. | .. | .. | 1 |
| STURGEON POACHER | .. | TRACE | TRACE | .. | .. | .. |
| OTHERS | .. | .. | .. | .. | .. | .. |
| INVERTEBRATES | | | | | | |
| OTHERS | 3 | 7 | .. | 44 | 20 | 42 |

Table 3d (cont'd)

APPENDIX TABLE 1 - WEIGHTS ( KG) PAGE 2.1

| SET NO. | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|
| DATE | JAN. 31 | FEB. 2 | FEB. 2 | FEB. 2 | FEB. 3 | FEB. 3 |
| AREA | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| TIME START (LST) | 1443 | 1221 | 1323 | 1447 | 823 | 1233 |
| DURATION(HR.MIN) | .30 | .30 | .30 | .30 | .30 | .30 |
| START N. LAT. (DEG) | 53 | 53 | 53 | 53 | 54 | 54 |
| (MIN) | 10.4 | 30.5 | 31.5 | 30.5 | 18.6 | 18.2 |
| W. LONG. (DEG) | 130 | 131 | 131 | 131 | 131 | 131 |
| (MIN) | 46.2 | 16.2 | 16.0 | 18.0 | 21.9 | 24.2 |
| DIRECTION (DEG.TRUE) | 344 | 308 | 296 | 360 | 259 | 250 |
| FINISH N. LAT. (DEG) | 53 | 53 | 53 | 53 | 54 | 54 |
| (MIN) | 12.1 | 31.4 | 32.1 | 31.4 | 18.4 | 17.8 |
| W.LONG. (DEG) | 130 | 131 | 131 | 131 | 131 | 131 |
| (MIN) | 46.8 | 18.0 | 18.2 | 18.0 | 24.4 | 26.4 |
| LENGTH OF SET KM. | 1.6 | 1.4 | 1.5 | 1.0 | 1.5 | 1.4 |
| DEPTH START (m-fa) | 141- 77 | 29- 15 | 26- 14 | 31- 16 | 97- 53 | 104- 56 |
| FINISH (m-fa) | 146- 79 | 28- 15 | 26- 14 | 24- 13 | 104- 56 | 106- 57 |
| INTERV. (m-fa) | 140- 70 | 20- 10 | 20- 10 | 20- 10 | 100- 50 | 100- 50 |
| SEE FIGURE NO. | | | | | | |
| START TEMP.(DEG.C) | .0 | .0 | .0 | .0 | .0 | .0 |
| END TEMP.(DEG.C) | .0 | .0 | .0 | .0 | .0 | .0 |
| TYPE OF GEAR | 0 | 0 | 0 | 0 | 0 | 0 |
| NO. OF GROUPS. | 0 | 0 | 0 | 0 | 0 | 0 |
| TOTAL CATCH (KG) | 209 | 4 | 4 | 10 | 701 | 330 |
| REMARKS | USABLE | USABLE | USABLE | USABLE | USABLE | USABLE |

Table 3d (cont'd)

APPENDIX TABLE 1 - WEIGHTS ( KG) PAGE 2.2

| SET NO. | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|
| DATE | JAN. 31 | FEB. 2 | FEB. 2 | FEB. 2 | FEB. 3 | FEB. 3 |
| AREA | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 |
| TOTAL CATCH WEIGHT | 225 | 4 | 4 | 10 | 701 | 330 |
| TOTAL CATCH PIECES | 184* | 3* | 4* | 1* | 1167* | 519* |
| GROUNDFISH | | | | | | |
| ENGLISH SOLE | 18 | .. | TRACE | .. | 14 | 19 |
| PIECES | 18* | .. | .. | .. | 14* | 19* |
| TURBOT | 19 | .. | .. | .. | 101 | 98 |
| PACIFIC COD | 44 | TRACE | TRACE | .. | 56 | 19 |
| DAB (PACIFIC) | .. | .. | .. | .. | TRACE | .. |
| LINGCOD | 18 | .. | .. | .. | .. | .. |
| HALIBUT | 17 | 1 | .. | 10 | 5 | 1 |
| PETRALE SOLE | 13 | .. | .. | .. | .. | .. |
| REX SOLE | 6 | .. | .. | .. | 2 | 1 |
| S. PAUCISPINIS | 17 | .. | .. | .. | .. | .. |
| ROCK SOLE | .. | 1 | 4 | TRACE | .. | .. |
| S. RUBERRIMUS | 5 | .. | .. | .. | .. | .. |
| S. PINNIGER | 2 | .. | .. | .. | .. | .. |
| STARRY FLOUNDER | .. | 2 | .. | .. | .. | .. |
| FLATHEAD SOLE | 1 | TRACE | .. | .. | .. | .. |
| WALLEYE POLLOCK | 1 | .. | .. | .. | TRACE | .. |
| OTHERS | .. | TRACE | TRACE | TRACE | TRACE | TRACE |
| SELACHI | | | | | | |
| RATFISH | 4 | .. | .. | .. | 471 | 191 |
| BIG SKATE | .. | .. | .. | .. | 52 | .. |
| DOGFISH | 28 | .. | .. | .. | .. | 1 |
| OTHERS | .. | .. | .. | .. | .. | .. |
| FISH | | | | | | |
| CYMATOGASTER | TRACE | .. | .. | .. | .. | .. |
| HERRING | TRACE | .. | .. | .. | .. | .. |
| STURGEON POACHER | .. | .. | .. | .. | .. | .. |
| OTHERS | .. | .. | .. | .. | .. | .. |
| INVERTEBRATES | | | | | | |
| OTHERS | 32 | TRACE | TRACE | .. | TRACE | TRACE |

## BRIDGE LOG

VESSEL:[1] _____ DATE:[4] _____ (yr. mo. day.) HAUL/SET NO.:[11] ___

LOCATION: major[13] _____ minor[15] _____ locality[17] _____

_____

GEAR:[19] _____ EFFECTIVE CODEND MESH (cm)[22] _____ (in) _____ CIRCLE:[25] codend — liner — other _____

START: lat[28] ° ' _____ long[31] ° ' _____ LORAN: _____

END: lat[36] ° ' _____ long[41] ° ' _____ LORAN: _____

TIME: (____) start[46] _____ finish _____ DURATION: (min)[50] _____

DIRECTION OF SET: (°T)[54] _____

DISTANCE TRAVELLED: (n. mi.)[57] _____ VESSEL SPEED: (kn)[60] _____

BOTTOM DEPTH: modal depth (m)[62] _____ start (m)[66] _____ finish (m)[70] _____
(fms) _____ (fms) _____

NET DEPTH: modal depth (m)[74] _____ /01/ start (m)[19] _____ finish (m)[23] _____
(headline to surface) (fms) _____ (fms) _____

MOUTH OPENING: (m)[27] _____ WARP LGTH: (m)[29] _____ SWEEP LGTH: (m) _____
(vertical) (fms) _____ (fms) _____ (fms) _____

FOOTROPE OFF BOTTOM: modal depth (m)[32] _____ start (m)[36] _____ finish (m)[40] _____
(fms) _____ (fms) _____

SOUNDER TYPE:[44] _____ TARGET DEPTH: (m)[46] _____ (fms) _____

SOUNDER SUMMARY:[50] _____

_____

WIND DIRECTION: (°T)[52] _____ WIND SPEED: (kn)[55] _____ WEATHER: _____

SEA STATE:[58] _____ TIDE:[60] _____ SWELL: _____

TEMP. (°C): surface[61] _____ bottom[64] _____ ASSOC. DEPTH: (m)[67] _____

OCEANOGRAPHIC DATA: (circle)[71] XBT - BT - STD - TDM - TTM - PT - ISK - CTD.

OTHER: specify _____

REASON FOR SET:[73] _____ RECORDER:[75] _____

REMARKS:[77] _____

_____ /02/

| 1 | | | 4 | | | | | | 10 | | | 13 | | 15 | | 17 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

COLUMNS 1-18 ARE THE SAME FOR CARD TYPES 01 & 02

| 19 | | | 22 | | | 25 | 26 | | | | | 31 | | | | | 36 | | | | | | 41 | | | | | | 46 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | | | | 54 | | | 57 | | | 60 | | 62 | | | | 66 | | | | 70 | | | | 74 | | | | | 78 | 79 | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | - | 0 | 1 |

| 19 | | | 23 | | | 27 | | 29 | | | 32 | | | | 36 | | | | 40 | | | | 44 | | 46 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | | 52 | | | 55 | | | 58 | | 60 | 61 | | | 64 | | | 67 | | | | 71 | | 73 | | 75 | | 77 | | 79 | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 | 2 |

HZ May/91

Fig. 1. Bridge log record sheet.

Haul number _____ Total lift 75 _____ lbs.

| ROCKFISH | | FLATFISH | | ROUNDFISH | |
|---|---|---|---|---|---|
| aleutianus | 394 | Butter sole | 619 | Blackcod | 455 |
| alutus | 396 | C-O sole | 633 | Hake | 225 |
| auriculatus | 398 | Curlfin sole | 635 | Lingcod | 467 |
| aurora | 400 | Dab (Pacific) | 596 | Pacific cod | 222 |
| babcocki | 401 | Dab (speckled) | 598 | Pac. pollock | 228 |
| borealis | 403 | Deepsea sole | 605 | Eelpout | 231 |
| brevispinis | 405 | Dover sole | 626 | Eulachon | 148 |
| caurinus | 407 | English sole | 628 | Hagfish | 016 |
| ciliatus | 409 | Flathead sole | 612 | Herring | 096 |
| crameri | 410 | Halibut ( ) | 614 | Longfin cod | 220 |
| diploproa | 412 | Petrale sole | 607 | Midshipman | 207 |
| elongatus | 414 | Rex sole | 610 | Poacher | 546 |
| emphaeus | 415 | Rock sole | 621 | Prowfish | 359 |
| entomelas | 417 | Sand sole | 636 | Ragfish | 385 |
| flavidus | 418 | Slender sole | 625 | Rattail | 249 |
| goodei | 420 | Starry flounder | 631 | Ronquils | 317 |
| helvomaculatus | 421 | Turbot | 602 | Sandfish | 316 |
| jordani | 423 | Yellowfin sole | 623 | Sandlance | 361 |
| maliger | 424 | | | Sculpins | 472 |
| melanops | 426 | | | Tomcod | 226 |
| miniatus | 428 | Total weight | | | |
| mystinus | 429 | SELACHII | | | |
| nebulosus | 431 | Cat shark | 038 | | |
| nigrocinctus | 433 | Dogfish | 044 | | |
| paucispinis | 435 | Ratfish | 066 | | |
| pinniger | 437 | Skates | 051 | | |
| polyspinis | 438 | | | | |
| proriger | 439 | Total weight | | Total weight | |
| reedi | 440 | INVERTEBRATES | | | |
| ruberrimus | 442 | Anemone: sm. lu. | 913 | Sponge: u.b.g | |
| saxicola | 444 | Coral | 914 | Squid: lol. sq.sm. | |
| variegatus | 446 | Crab | 960 | Starfish: | 922 |
| wilsoni | 448 | Cucumber | 926 | basket;brittle | 923 |
| zacentrus | 450 | Octopus | 935 | Urchin:rd;pu;gr. | 924 |
| alascanus | 451 | Shrimp: pink | 956 | | |
| | | prawn | 957 | | |
| Total weight | | Side st. other | | Total weight | |

Fig. 2. Catch log record sheet.

Appendix table A. Fish Code File.

| | | |
|---|---|---|
| 00001 | 222 | PACIFIC COD |
| 00001 | 228 | WALLEYE POLLOCK |
| 00001 | 418 | S. FLAVIDUS |
| 00001 | 435 | S. PAUCISPINIS |
| 00001 | 437 | S. PINNIGER |
| 00001 | 442 | S. RUBERRIMUS |
| 00001 | 446 | S. VARIEGATUS |
| 00001 | 467 | LINGCOD |
| 00001 | 596 | DAB (PACIFIC) |
| 00001 | 598 | DAB (SPECKLED) |
| 00001 | 602 | TURBOT |
| 00001 | 607 | PETRALE SOLE |
| 00001 | 610 | REX SOLE |
| 00001 | 612 | FLATHEAD SOLE |
| 00001 | 614 | HALIBUT |
| 00001 | 619 | BUTTER SOLE |
| 00001 | 621 | ROCK SOLE |
| 00001 | 625 | SLENDER SOLE |
| 00001 | 626 | DOVER SOLE |
| 00001 | 628 | ENGLISH SOLE |
| 00001 | 631 | STARRY FLOUNDER |
| 00001 | 635 | CURLFIN SOLE |
| 00001 | 636 | SAND SOLE |
| 00002 | 044 | DOGFISH |
| 00002 | 056 | BIG SKATE |
| 00002 | 066 | RATFISH |
| 00003 | 096 | HERRING |
| 00003 | 304 | CYMATOGASTER |
| 00003 | 550 | STURGEON POACHER |
| 00004 | 3L0 | ANEMONE |
| 00004 | 4GA | STARFISH |
| 00004 | 5QA | BASKET STARS |
| 00004 | 6AB | SEA URCHINS |
| 00004 | 92A | SQUID |
| 00004 | 999 | UNKNOWN INVERTEBRATES |
| 00004 | SAB | SHRIMP |
| 00004 | SQA | EUALUS SPP. |
| 00004 | VAA | ANOMURA |
| 00004 | VAC | HERMIT CRAB |

Appendix table B. Estimated Average Weight File.

| | |
|---|---|
| 044 | 2.50 |
| 066 | 0.50 |
| 096 | 0.12 |
| 467 | 3.00 |
| 614 | 10.00 |

Appendix table C. Group Name File.

GROUNDFISH
SELACHI
FISH
INVERTEBRATES
OTHERS
OTHERS
OTHERS
OTHERS
OTHERS
OTHERS

Appendix table D. Line Title File.

SET NO.
DATE
AREA
TIME START (LST)
DURATION(HR.MIN)
START N. LAT. (DEG)
(MIN)
W. LONG. (DEG)
(MIN)
DIRECTION (DEG.TRUE)
FINISH N. LAT. (DEG)
(MIN)
W.LONG. (DEG)
(MIN)
LENGTH OF SET KM.
DEPTH (m-fa)
SEE FIGURE NO.
START TEMP.(DEG.C)
END TEMP.(DEG.C)
TYPE OF GEAR
NO. OF GROUPS.
TOTAL CATCH (KG)
REMARKS

Appendix table E. Remarks File.

```
  USABLE
UNUSABLE
    SNAG    USABLE
    SNAG  UNUSABLE
 X-DOORS  UNUSABLE
NET TORN    USABLE
NET TORN  UNUSABLE
ESTIMATE
DUMPED    ALL JUNK
SP.NOTED  NO WTS
GEAR LOST UNUSABLE
```

Appendix table F. INIT.DAT File.

```
SPEC2.DAT
FISHAVE.DAT
FLATFISH.DAT
GROUP.DAT
LINETITLE.DAT
REMARKS.DAT
TEST.DAT
GROUNDFISH 1986 ASSEMBLAGE SURVEY
R/V G.B. REED
J. FARGO
 15 10  5  0100100100100100100
W
100000
```

Appendix table G. Trip Data, 1986 Assemblage Survey - R/V G.B. Reed.

```
R01860129  1                5305730529530683055012l2  30309 1632  86  88  84   01
R01860129  1      415202 122442 415218 122278         085 20  1               02
R01860129  1        197  92A        2                                         03
R01860129  1        197  614        4                                         03
R01860129  1        197  628      116                                         03
R01860129  1        197  222        6                                         03
R01860129  1        197  596       45                                         03
R01860129  1        197  621        1                                         03
R01860129  1        197  602       -1                                         03
R01860129  1        197  610       -1                                         03
R01860129  1        197  066        1                                         03
R01860129  1        197  056        4                                         03
R01860129  1                5305730529530683055012l2  30309 1632  86  88  84   01
R01860129  1      415202 122442 415218 122278         085 20  1               02
R01860129  1        197  92A        2                                         03
R01860129  1        197  614        4                                         03
R01860129  1        197  628      116                                         03
R01860129  1        197  222        6                                         03
R01860129  1        197  596       45                                         03
R01860129  1        197  621        1                                         03
R01860129  1        197  602       -1                                         03
R01860129  1        197  610       -1                                         03
R01860129  1        197  066        1                                         03
R01860129  1        197  056        4                                         03
R01860129  1        197  228       -1                                         03
R01860129  1        197  3L0       -1                                         03
R01860129  1        197  467       18                                         03
R01860129  1        197  999        2                                         03
R01860129  2                5305230533530623055713330 30306 1836  77  79  75   01
R01860129  2      415182 122448 415192 122280         085 15  1               02
R01860129  2        172  304       -1                                         03
R01860129  2        172  VAA       -1                                         03
R01860129  2        172  614       13                                         03
R01860129  2        172  628       35                                         03
R01860129  2        172  619       -1                                         03
R01860129  2        172  550       -1                                         03
R01860129  2        172  610       -1                                         03
R01860129  2        172  92A        4                                         03
R01860129  2        172  222       10                                         03
R01860129  2        172  467        8                                         03
R01860129  2        172  602       -1                                         03
R01860129  2        172  056       57                                         03
R01860129  2        172  621        1                                         03
R01860129  2        172  636       -1                                         03
R01860129  2        172  596       44                                         03
R01860129  2        172  999        4                                         03
R01860130  3                5305130532530613054908 28 30312 1530  77  77  77   01
R01860130  3      415177 122459 415198 122311         085 30  2               02
R01860130  3        277  550       -1                                         03
R01860130  3        277  92A        1                                         03
R01860130  3        277  304        1                                         03
```

Appendix table G. (cont'd)

```
R01860130  3        277  4GA       -1                                                    03
R01860130  3        277  222       12                                                    03
R01860130  3        277  3L0       -1                                                    03
R01860130  3        277  619       -1                                                    03
R01860130  3        277  635       -1                                                    03
R01860130  3        277  621        3                                                    03
R01860130  3        277  056       41                                                    03
R01860130  3        277  596       81                                                    03
R01860130  3        277  628      138                                                    03
R01860130  3        277  467       -1                                                    03
R01860130  3        277  607       -1                                                    03
R01860130  3        277  999        1                                                    03
R01860130  4             53109304735312630485134O  30337 1836 127 130 124               01
R01860130  4     415460 122279 415505 122110       110 24  1                            02
R01860130  4        227  222       13                                                    03
R01860130  4        227  610       14                                                    03
R01860130  4        227  4GA       -1                                                    03
R01860130  4        227  228                                                             03
R01860130  4        227  304        2                                                    03
R01860130  4        227  066        1                                                    03
R01860130  4        227  467        9                                                    03
R01860130  4        227  044       57                                                    03
R01860130  4        227  626       -1                                                    03
R01860130  4        227  6AB       -1                                                    03
R01860130  4        227  92A       21                                                    03
R01860130  4        227  614        5                                                    03
R01860130  4        227  628       60                                                    03
R01860130  4        227  602       16                                                    03
R01860130  4        227  607       27                                                    03
R01860130  4        227  5QA        2                                                    03
R01860130  4        227  999       23                                                    03
R01860130  5             53102304765311830487151O  30336 1836 129 130 128               01
R01860130  5     415427 122323 415475 122160       110 28  1                            02
R01860130  5         91  92A       10                                                    03
R01860130  5         91  044       26                                                    03
R01860130  5         91  607        1                                                    03
R01860130  5         91  614       11                                                    03
R01860130  5         91  222       11                                                    03
R01860130  5         91  602        2                                                    03
R01860130  5         91  467       13                                                    03
R01860130  5         91  066       -1                                                    03
R01860130  5         91  304        1                                                    03
R01860130  5         91  628       11                                                    03
R01860130  5         91  610        5                                                    03
R01860130  5         91  999       10                                                    03
R01860131  6             53097304795311030487082O  30340 1428 128 130 126               01
R01860131  6     415409 122334 415446 122212       140 22  2                            02
R01860131  6        165  044       26                                                    03
R01860131  6        165  096        1                                                    03
R01860131  6        165  607        9                                                    03
R01860131  6        165  SQA        3                                                    03
```

Appendix table G (cont'd)

```
R01860131  6          165  628        34                                                    03
R01860131  6          165  610        20                                                    03
R01860131  6          165  4GA         2                                                    03
R01860131  6          165  6AB         2                                                    03
R01860131  6          165  92A        14                                                    03
R01860131  6          165  056        14                                                    03
R01860131  6          165  228        -1                                                    03
R01860131  6          165  467         2                                                    03
R01860131  6          165  435         2                                                    03
R01860131  6          165  222        28                                                    03
R01860131  6          165  066        -1                                                    03
R01860131  6          165  304         1                                                    03
R01860131  6          165  625        -1                                                    03
R01860131  6          165  602         7                                                    03
R01860131  6          165  446        -1                                                    03
R01860131  6          165  999        21                                                    03
R01860131  7               5310430462531213046814 43  30344 1632 143 141 146              01
R01860131  7     415452 122366 415507 122218           120 28  1                          02
R01860131  7          209  467        18                                                    03
R01860131  7          209  228         1                                                    03
R01860131  7          209  612         1                                                    03
R01860131  7          209  437         2                                                    03
R01860131  7          209  92A        14                                                    03
R01860131  7          209  614        17                                                    03
R01860131  7          209  222        44                                                    03
R01860131  7          209  096        -1                                                    03
R01860131  7          209  066         4                                                    03
R01860131  7          209  607        13                                                    03
R01860131  7          209  5QA         2                                                    03
R01860131  7          209  628        18                                                    03
R01860131  7          209  304        -1                                                    03
R01860131  7          209  610         6                                                    03
R01860131  7          209  435        17                                                    03
R01860131  7          209  044        28                                                    03
R01860131  7          209  442         5                                                    03
R01860131  7          209  602        19                                                    03
R01860131  7          209  999        16                                                    03
R01860202  8               5330531162533143118012 21  30308 1428  27  29  28             01
R01860202  8     415798 119717 415804 119585           060  8  2                          02
R01860202  8            4  614         1                                                   03
R01860202  8            4  92A        -1                                                   03
R01860202  8            4  631         2                                                   03
R01860202  8            4  598        -1                                                   03
R01860202  8            4  222        -1                                                   03
R01860202  8            4  5QA        -1                                                   03
R01860202  8            4  621         1                                                   03
R01860202  8            4  612        -1                                                   03
R01860202  8            4  999        -1                                                   03
R01860202  9               5331531160533213118213 23  30296 1530  26  26  26             01
R01860202  9     415830 119649 415824 119526           060 11  0                          02
R01860202  9            4  636        -1                                                   03
```

Appendix table G (cont'd)

```
R01860202  9          4  628       -1                                                    03
R01860202  9          4  598       -1                                                    03
R01860202  9          4  222       -1                                                    03
R01860202  9          4  VAC       -1                                                    03
R01860202  9          4  621        4                                                    03
R01860202  9          4  999       -1                                                    03
R01860202 10             533053118053314311801447  30360 1020  28  31  24              01
R01860202 10     415775 119658 415783 119513       060  8  1                            02
R01860202 10         10  636       -1                                                    03
R01860202 10         10  614       10                                                    03
R01860202 10         10  621       -1                                                    03
R01860203 11             541863121954184312440823  30259 1530 101  97 104              01
R01860203 11     417230 115700 417190 115640       010 12  2                            02
R01860203 11        701  056       52                                                    03
R01860203 11        701  066      471                                                    03
R01860203 11        701  418       -1                                                    03
R01860203 11        701  596       -1                                                    03
R01860203 11        701  228       -1                                                    03
R01860203 11        701  610        2                                                    03
R01860203 11        701  222       56                                                    03
R01860203 11        701  SAB       -1                                                    03
R01860203 11        701  628       14                                                    03
R01860203 11        701  4GA       -1                                                    03
R01860203 11        701  614        5                                                    03
R01860203 11        701  602      101                                                    03
R01860203 11        701  999       -1                                                    03
R01860203 12             541823124254178312641233  30250 1428 105 104 106              01
R01860203 12     417186 115652 417146 115624       020 12  2                            02
R01860203 12        330  222       19                                                    03
R01860203 12        330  066      191                                                    03
R01860203 12        330  628       19                                                    03
R01860203 12        330  619       -1                                                    03
R01860203 12        330  602       98                                                    03
R01860203 12        330  614        1                                                    03
R01860203 12        330  044        1                                                    03
R01860203 12        330  610        1                                                    03
R01860203 12        330  SAB       -1                                                    03
R01860203 12        330  999       -1                                                    03
```

Appendix: Program Listing

LIABILITY DISCLAIMER FOR SOFTWARE SHARING
==========

The Government of Canada shall have no liability or responsibility to any person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by this software, including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of this software.

There is no service associated with this software and any defect or fault found therein will not be the responsibility of the Government of Canada.

FILE: GROUND.FOR

```
C234567
        LOGICAL REP(6)
        CHARACTER*80 FILENAME(7)
        CALL INIT(FILENAME,7,REP)
        CALL MMAIN(FILENAME,7,REP)
      STOP
      END
      SUBROUTINE GETINT(NUM)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE GETS AN INTEGER FROM THE KEYBOARD
C
C IMPORTS : NONE
C EXPORTS : NUM
C-------------------------------------------------------------------------------

        INTEGER NUM,MULT,I
        CHARACTER LINE(20)*1
        READ(*,FMT='(20A1)') (LINE(I),I=1,20)
        I = 1
10      IF ((LINE(I).LT.'0' .OR. LINE(I).GT.'9') .AND. LINE(I).NE.'-'
     I      .AND. I.LT.20) THEN
              I = I + 1
              GOTO 10
        ENDIF
        MULT = 1
        IF (LINE(I).EQ.'-') THEN
           MULT = -1
           I = I + 1
        ENDIF
        NUM = 0
20      IF (LINE(I).GE.'0' .AND. LINE(I).LE.'9' .AND. I.LE.20) THEN
              NUM = NUM * 10 + ICHAR(LINE(I))- ICHAR('0')
              I = I + 1
              GOTO 20
```

```
         ENDIF
         NUM = NUM * MULT
      RETURN
      END
      SUBROUTINE MAINMENU(LIST,NUM)
         CHARACTER*30 LIST(NUM)
         PRINT 9
         PRINT 12,(LIST(I)(1:1),LIST(I)(2:),I=1,NUM)
 12      FORMAT(1X,'<',A1,'>',A29)
  9      FORMAT(///,T20,'MAIN MENU')
      RETURN
      END
      SUBROUTINE MMAIN(FILENAME,NUM,REP)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE IS THE MAIN DRIVER FOR THE PROGRAM
C
C IMPORTS : FILENAME,NUM,REP
C EXPORTS : FILENAME,REP
C
C PROCEDURES CALLED : MAINMENU,FOPTION,GOPTION,POPTION,ROPTION,SOPTION
C                     WOPTION
C-------------------------------------------------------------------------------
         INTEGER NUM
         CHARACTER*80 FILENAME(NUM)
         CHARACTER*30 MAINLIST(7)
         CHARACTER*1 CHOICE
         LOGICAL REP(6),CHANGE/.FALSE./
         DATA MAINLIST/'FILE OPTIONS','GROUPING OPTIONS',
     D                 'PAGE HEADER ',
     D                 'REPORTS TO BE GENERATED','START',
     A                 'WRITE OPTIONS TO DISK','QUIT'/

         CALL MAINMENU(MAINLIST,7)
         PRINT 12
         READ (*,11) CHOICE
 11      FORMAT(A1)
 10      IF (CHOICE.NE.'Q') THEN
            IF (CHOICE.EQ.'F') THEN
               CHANGE = .TRUE.
               CALL FOPTION(FILENAME,NUM)
            ELSEIF (CHOICE.EQ.'G') THEN
               CHANGE = .TRUE.
               CALL GOPTION
            ELSEIF (CHOICE.EQ.'P') THEN
               CHANGE = .TRUE.
               CALL POPTION
            ELSEIF (CHOICE.EQ.'R') THEN
               CHANGE = .TRUE.
               CALL ROPTION(REP)
            ELSEIF (CHOICE.EQ.'S') THEN
               CALL SOPTION(FILENAME,NUM,REP)
            ELSEIF (CHOICE.EQ.'W') THEN
               CHANGE = .FALSE.
               CALL WOPTION(FILENAME,NUM,REP)
```

```
            ELSE
               PRINT 13
            ENDIF
            CALL MAINMENU(MAINLIST,7)
            PRINT 12
            READ (*,11) CHOICE
            GOTO 10
         ENDIF
 12      FORMAT(' PLEASE ENTER YOUR CHOICE ( F,G,P,R,S,W OR Q )')
 13      FORMAT(' INVALID CHOICE PLEASE ENTER A VALID CHOICE')
 14      FORMAT(' NEW OPTIONS NOT SAVED. DO YOU WISH TO SAVE?(Y/N)')
         IF (CHANGE) THEN
            PRINT 14
            READ(*,FMT='(A1)') CHOICE
            IF (CHOICE.EQ.'Y') CALL WOPTION(FILENAME,NUM,REP)
         ENDIF
      RETURN
      END
      SUBROUTINE FMENU(LIST1,LIST2,NUM)
         INTEGER NUM
         CHARACTER*80 LIST1(NUM),LIST2(NUM)
         PRINT 9
         PRINT '(1X,I2,''.'',A70/4X,A70)',(I,LIST1(I),LIST2(I),I=1,NUM)
 9       FORMAT(///,T20,'FILE MENU')
      RETURN
      END
      SUBROUTINE FOPTION(LIST,NUM)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE IS THE MAIN DRIVER FOR THE 'FILE' OPTION
C
C IMPORTS : LIST,NUM
C EXPORTS : LIST
C
C PROCEDURES CALLED : FMENU,GETINT
C-------------------------------------------------------------------------------
         INTEGER NUM,CHOICE,LENG
         CHARACTER*1 ACHOICE
         CHARACTER*80 LIST(NUM),LIST2(7),NAME
         DATA LIST2/'FISH CODE FILE','ESTIMATED AVERAGES FILE',
     D              'TRIP DATA FILE',
     D              'GROUP NAME FILE','PAGE 1 LINE TITLE FILE',
     A              'CATCH REMARKS FILE','OUTPUT FILE'/

         CALL FMENU(LIST2,LIST,NUM)
         PRINT 12
         CALL GETINT(CHOICE)
 12      FORMAT (' ENTER YOUR CHOICE (DEFAULT = 0 TO QUIT)')
 10      IF (CHOICE.LE.0 .OR. CHOICE.GT.NUM) GOTO 100
            PRINT 11,LIST2(CHOICE)
 11         FORMAT(' PLEASE ENTER ',A60)
            READ(*,'(A)') NAME
            LENG = INDEX(NAME,' ')-1
            IF (LENG.GT.0) LIST(CHOICE) = NAME
            CALL FMENU(LIST2,LIST,NUM)
```

```
            PRINT 12
            CALL GETINT(CHOICE)
         GOTO 10
100   RETURN
      END
      SUBROUTINE GMENU(LIST1,WORP)
         INTEGER LIST1(10)
         CHARACTER*1 WORP
         PRINT 9
         PRINT 11
         PRINT 12,(I,LIST1(I),I=1,10)
         PRINT 14,WORP
 9       FORMAT(///,T20,'GROUP MENU')
11       FORMAT(1X,'GROUP ','    MAX NO./GRP')
12       FORMAT(1X,I3,3X,I10)
14       FORMAT(1X,' 11 <W>EIGHTS OR <P>IECES = ',A1)
      RETURN
      END
      SUBROUTINE GOPTION
C-------------------------------------------------------------------------------
C THIS SUBROUTINE IS THE DRIVER FOR THE 'GROUPING' OPTION.
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : GROUP
C                    VARIABLES : GMAX
C                       : CATCH
C                             : CWORP
C
C PROCEDURES CALLED : GMENU,GETINT
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
        INTEGER CHOICE
        CHARACTER CH*1
        CALL GMENU(GMAX,CWORP)
        PRINT 12
        CALL GETINT(CHOICE)
10      IF (CHOICE.GT.0 .AND. CHOICE.LE.11) THEN
           IF (CHOICE.LE.10) THEN
              PRINT 11,CHOICE
              CALL GETINT(GMAX(CHOICE))
           ELSE
              PRINT 14
              READ(*,'(A1)') CH
              IF (CH.EQ.'P'.OR.CH.EQ.'W') CWORP = CH
           ENDIF
           CALL GMENU(GMAX,CWORP)
           PRINT 12
           CALL GETINT(CHOICE)
           GOTO 10
        ENDIF
 9      FORMAT(///,T20,'GROUP MENU')
11      FORMAT (1X,'ENTER NEW MAX NO/GROUP ',I2)
12      FORMAT (1X,'ENTER n TO CHANGE MAX/GROUP , 11 TO CHANGE ',
```

```
     0           '<W>EIGHTS OR <P>IECES ,0 TO QUIT')
 14      FORMAT (1X,'ENTER <W>EIGHT OR <P>IECES')
       RETURN
       END

       SUBROUTINE PMENU(LIST1,NUM)
          INTEGER NUM
          CHARACTER*80 LIST1(NUM)
          PRINT 9
          PRINT '(1X,I2,''.'',A74)',(I,LIST1(I),I=1,NUM)
  9       FORMAT(///,T20,'PAGE HEADER MENU')
       RETURN
       END
       SUBROUTINE POPTION
C-------------------------------------------------------------------------------
C THIS SUBROUTINE IS THE DRIVER FOR THE 'PAGE HEADER' OPTION
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : OUTPUT
C                    VARIABLES : PHEADER
C
C PROCEDURES CALLED : GETINT,PMENU
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
         INTEGER CHOICE
         CHARACTER*80 LINE

         CALL PMENU(PHEADER,3)
         PRINT 12
         CALL GETINT(CHOICE)
 12      FORMAT (' ENTER YOUR CHOICE (DEFAULT = 0 TO QUIT)')
 10      IF (CHOICE.LE.0 .OR. CHOICE.GT.3) GOTO 100
            PRINT 11
 11         FORMAT(' PLEASE ENTER NEW PAGE HEADER LINE')
            READ(*,'(A)') LINE
            PHEADER(CHOICE) = LINE
            CALL PMENU(PHEADER,3)
            PRINT 12
            CALL GETINT(CHOICE)
         GOTO 10
 100  RETURN
      END
      SUBROUTINE RMENU(REP,REPTITLE)
         CHARACTER*30 REPTITLE(6)
         LOGICAL REP(6)
         PRINT 9
         DO 10 I = 1,6,1
            IF (REP(I)) THEN
               PRINT 11,'ON ',I,REPTITLE(I)
            ELSE
               PRINT 11,'OFF',I,REPTITLE(I)
            ENDIF
 10      CONTINUE
```

```
  9      FORMAT(///,T20,' REPORT MENU')
 11      FORMAT(1X,A3,I2,'.',A30)
      RETURN
      END
      SUBROUTINE ROPTION(REP)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE OS THE DRIVER FOR THE 'REPORTS' OPTION.
C
C IMPORTS : REP
C EXPORTS : REP
C COMMON BLOCK ACCESSED : OUTPUT
C                     VARIABLES : REPTITLE
C
C PROCEDURES CALLED : RMENU
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
        LOGICAL REP(6)
        INTEGER CHOICE
        CALL RMENU(REP,REPTITLE)
        PRINT 12
        CALL GETINT(CHOICE)
 10     IF (CHOICE.NE.0 .AND. CHOICE.LE.6) THEN
           REP(CHOICE) = (.NOT.REP(CHOICE))
           CALL RMENU(REP,REPTITLE)
           PRINT 12
           CALL GETINT(CHOICE)
           GOTO 10
        ENDIF
 12     FORMAT (1X,'ENTER n TO TOGGLE ON/OFF, 0 TO QUIT')
      RETURN
      END
      SUBROUTINE SOPTION(FILENAME,NUM,REP)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE IS THE DRIVER FOR THE 'START' OPTION.
C
C IMPORTS : FILENAME,NUM,REP
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : OUTPUT
C                     VARIABLES : OUTF
C
C PROCEDURES CALLED : FISHINIT,AVEINIT,GROUPINIT,LINEINIT,REMINIT,DATAINIT,
C                      PROCDATA,SUMMARY,ESTIMATE,SUMMARY,REPORT
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
        CHARACTER*80 FILENAME(NUM)
        LOGICAL REP(6),DEBUG/.FALSE./

        IF (DEBUG) PRINT*,'OPEN'
C OPEN OUTPUT FILE
        OPEN(UNIT=OUTF,FILE=FILENAME(7),STATUS='UNKNOWN',ERR=100)

        IF (DEBUG) PRINT *,'FISHINIT'
C INITIALIZE FISH LIST
        CALL FISHINIT(FILENAME(1))
```

```
          IF (DEBUG) PRINT *,'AVEINIT'
C INITIALIZE AVERAGE LIST
          CALL AVEINIT(FILENAME(2))

          IF(DEBUG) PRINT*,'GROUPINIT'
C INITIALIZE GROUP NAME LIST
          CALL GROUPINIT(FILENAME(4))

          IF (DEBUG) PRINT*,'LINEINIT'
C INITLIALIZE LINE TITLE LIST
          CALL LINEINIT(FILENAME(5))

          IF (DEBUG) PRINT*,'REMINIT'
C INITIALIZE REMARKS LIST
          CALL REMINIT(FILENAME(6))

          IF (DEBUG) PRINT *,'DATAINIT'
C READ IN DATA
          CALL DATAINIT(FILENAME(3))

          IF (DEBUG) PRINT*,'PROCDATA'
C PROCESS DATA
          CALL PROCDATA
          IF(DEBUG) PRINT*,'SUMMARY'
C PRINT SUMMARY
          CALL SUMMARY

          IF (DEBUG) PRINT*,'ESTIMATE'
C ESTIMATE VALUES NOT GIVEN
          CALL ESTIMATE
          IF (DEBUG) PRINT*,'ESUMMARY'
C PRINT SUMMARY AFTER ESTIMATES DONE
          CALL ESUMMARY

          IF (DEBUG) PRINT*,'QUALIFY'
C DETERMINE SPECIES TO QUALIFY
          CALL QUALIFY
          IF(DEBUG) PRINT*,'PRQUALIFY'
C PRINT LIST OF SPECIES TO QUALIFY
          CALL PRQUALIFY
          IF (DEBUG) PRINT*,'REPORT'
C GENERATE REQUESTED REPORTS
          DO 10 I = 1,6,1
             IF (REP(I)) CALL REPORT(I,HNUM)
  10      CONTINUE
          IF (DEBUG) PRINT*,'CLOSE'
C CLOSE OUTPUT FILE
          CLOSE(OUTF)
       RETURN
  100  PRINT*, 'ERROR OPENING ',FILENAME(7)
       RETURN
       END
       SUBROUTINE INIT(FILENAME,NUM,REP)
C-------------------------------------------------------------------------------
```

```
C THIS SUBROUTINE DOES THE INITIALIZATIONS REQUIRED FOR PROGRAMME TO FUNCTION
C CORRECTLY.
C
C IMPORTS : NUM
C EXPORTS : FILENAME,REP
C
C COMMON BLOCK ACCESSED : OUTPUT
C                     VARIABLE : REPTITLE,OUTF,PHEADER
C                        : GROUP
C                              : GMAX
C                        : CATCH
C                              : CWORP
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
        LOGICAL REP(6)
        INTEGER IREP(6)
        INTEGER NUM
        REAL AMINI(10)
        CHARACTER*80 FILENAME(NUM)

        OUTF=99
        WTUNIT = ' KG'

C INITIALIZE REPORT TITLES
        REPTITLE(1)=' WEIGHTS ( KG)'
        REPTITLE(2)=' WEIGHTS ( KG)/( KM)'
        REPTITLE(3)=' WEIGHTS ( KG)/( KM*HRS)'
        REPTITLE(4)=' PIECES '
        REPTITLE(5)=' PIECES / KM'
        REPTITLE(6)=' PIECES /(KM*HRS)'

C OPEN 'INIT.DAT' WHICH CONTAINS INITIAL SETTINGS FOR PROGRAMME
        OPEN(UNIT=10,FILE='INIT.DAT',STATUS='OLD')

C READ IN LIST OF FILENAMES
        DO 20 I = 1,NUM,1
           READ(10,25,END=30) FILENAME(I)
  20    CONTINUE
  25    FORMAT(A)

C READ IN PAGE HEADER LINES
        DO 10 I = 1,3,1
           READ(10,'(A)',END=30) PHEADER(I)
  10    CONTINUE

C READ IN MAXIMUM NUMBER PER GROUP TO PRINT
        READ(10,FMT='(10I3)',END=30) (GMAX(I),I=1,10)
        GMAX(11) = 0

C READ IN SORT KEY ('W' FOR WEIGHTS OR 'P' FOR PIECES)
        READ(10,'(A1)',END=30) CWORP

C READ IN LIST OF REPORTS TO BE GENERATED
        READ(10,FMT='(6I1)',END=30) (IREP(I),I=1,6)
```

```
          DO 60 I = 1,6
             IF (IREP(I).GT.0 .AND. IREP(I).LE.6 ) REP(IREP(I))=.TRUE.
   60     CONTINUE

 C CLOSE 'INIT.DAT'
   30     CLOSE(10)
       RETURN
       END
       SUBROUTINE WOPTION(FILENAME,NUM,REP)
 C-------------------------------------------------------------------------------
 C
 C IMPORTS : NUM,FILENAME,REP
 C EXPORTS : NONE
 C
 C COMMON BLOCK ACCESSED : OUTPUT
 C                     VARIABLE : OUTF,PHEADER
 C                        : GROUP
 C                              : GMAX
 C                        : CATCH
 C                              : CWORP
 C-------------------------------------------------------------------------------
 $INCLUDE:'COMMON.INC'
          LOGICAL REP(6)
          INTEGER IREP(6)
          INTEGER NUM
          CHARACTER*80 FILENAME(NUM)

 C OPEN 'INIT.DAT' AND WRITE NEW OPTIONS
          OPEN(UNIT=10,FILE='INIT.DAT',STATUS='OLD')

 C WRITE LIST OF FILENAMES
          DO 20 I = 1,NUM,1
             WRITE(10,25) FILENAME(I)
   20     CONTINUE
   25     FORMAT(A)

 C WRITE PAGE HEADER
          DO 10 I = 1,3,1
             WRITE(10,FMT='(A)') PHEADER(I)
   10     CONTINUE

 C WRITE MAXIMUM NUMBER PER GROUP
          WRITE(10,FMT='(10I3)') (GMAX(I),I=1,10)

 C WRITE SORT KEY
          WRITE(10,'(A1)') CWORP

 C WRITE NUMBERS OF REPORTS TO BE GENERATED
          DO 60 I = 1,6
             IREP(I) = 0
             IF (REP(I)) IREP(I)=I
   60     CONTINUE
          WRITE(10,FMT='(6I1)') (IREP(I),I=1,6)
```

```
C CLOSE 'INIT.DAT'
  30     CLOSE(10)
      RETURN
      END
```

FILE: COMMON.INC

```
      CHARACTER*1  WPEST(300,101,2),CWORP
      COMMON/WPESTIMATE/WPEST
      CHARACTER PAGE1(66)*80,PAGE2(132)*80,PHEADER(3)*80,
     C          LINETITLE(30)*20,REMARKS(30)*20,REPTITLE(6)*30,
     H          FNAME(300)*16,CNAME(300)*16,GNAME(11)*16,
     A          FCODE(300)*3,ACODE(300)*3,CCODE(300)*3,
     R          WTUNIT*3,DUNIT*3,TUNIT*3
      REAL AWEIGHT(300),CEST(300),CWEIGHT(300,101)
      INTEGER FNUM,ANUM,CNUM,GNUM,HNUM,LNUM,REMNUM,GMAX(11)
      INTEGER FGROUP(300),CGROUP(300),OUTF,CPIECE(300,101)
      LOGICAL CGOTHERS(300)
      COMMON/UNITS/WTUNIT,DUNIT,TUNIT
      COMMON/LISTSIZES/FNUM,HNUM,CNUM,GNUM,ANUM,LNUM,REMNUM
      COMMON/FISH/FNAME,FCODE,FGROUP
      COMMON/AVEWEIGHT/ACODE,AWEIGHT
      COMMON/CATCH/CPIECE,CWEIGHT,CCODE,CNAME,CGROUP,CEST,CGOTHERS,
     C             CWORP
      COMMON/GROUP/GNAME,GMAX
      COMMON/OUTPUT/PAGE1,PAGE2,OUTF,PHEADER,LINETITLE,
     C              REMARKS,REPTITLE
C-------------------------------------------------------------------------------
C  COMMON BLOCK : WPESTIMATE
C      VARIABLE : WPEST(I,J,1) DETERMINES IF Ith SPECIES, Jth HAUL , NUMBER OF
C                      PIECES WAS ESTIMATED.
C                 WPEST(I,J,2) DETERMINES IF Ith SPECIES, Jth HAUL , WEIGHT WAS
C                      ESTIMATED.
C
C  COMMON BLOCK : UNITS
C      VARIABLE : WTUNIT = WEIGHT UNITS
C                 DUNIT = DEPTH UNITS
C                 TUNIT = TIME ZONE
C
C  COMMON BLOCK : LISTSIZES
C      VARIABLE : FNUM = NUMBER OF FISH CODES READ IN FROM FISH CODE FILE
C                 ANUM = NUMBER OF EST. AVE. WTS. READ IN FROM FILE
C                 HNUM = NUMBER OF HAULS ON TRIP
C                 CNUM = NUMBER OF SPECIES CAUGHT ON TRIP
C                 GNUM = NUMBER OF GROUP NAMES READ IN
C                 LNUM = NUMBER OF LINE TITLES READ IN
C                 REMNUM = NUMBER OF REMARKS READ IN
C
C  COMMON BLOCK : FISH
C      VARIABLE : FNAME = LIST OF FISH NAMES
C                 FCODE = LIST OF CORRESPONDING FISH CODES
C                 FGROUP = LIST OF CORRESPONDING FISH GROUP NUMBERS
C
C  COMMON BLOCK : AVEWEIGHT
C      VARIABLE : ACODE = LIST OF FISH CODES
C                 AWEIGHT = CORRESPONDING LIST OF ESTIMATED AVERAGE WEIGHTS.
```

```
C
C  COMMON BLOCK : CATCH
C      VARIABLE : CPIECE(I,J) = # OF PIECES OF THE Ith SPECIES FOR THE Jth HAUL.
C                 CWEIGHT(I,J) = WEIGHT OF THE Ith SPECIES FOR THE Jth HAUL.
C                 CCODE = LIST OF SPECIES CODES OF FISH CAUGHT.
C                 CNAME = CORRESPONDING LIST OF FISH NAMES.
C                 CGROUP = CORRESPONDING LIST OF GROUP NUMBERS.
C                 CEST = CORRESPONDING LIST OF EST. AVE. WTS.
C                 CGOTHERS = BOOLEAN LIST TO DETERMINE IF SPECIES SHOULD BE
C                            PRINTED.
C                 CWORP = DETERMINES IF GROUPS ARE TO BE SORTED IN DECREASING
C                         ORDER BY 'W'EIGHTS OR 'P'IECES.
C
C  COMMON BLOCK : GROUP
C      VARIABLE : GNAME = LIST OF GROUP NAMES.
C                 GMAX = MAXIMUM NUMBER PER GROUP TO PRINTED
C
C  COMMON BLOCK : OUTPUT
C      VARIABLE : PAGE1 = PAGE 1 OF TWO PAGE REPORT
C                 PAGE2 = PAGE 2 OF TWO PAGE REPORT
C                 OUTF = LOGICAL UNIT NUMBER FOR OUTPUT FILE
C                 PHEADER = HEADER FOR PAGES 1.1 AND 1.2 OF EACH REPORT
C                 LINETITLE = LINE TITLES FOR PAGE 1 OF REPORT
C                 REMARKS = CORRESPONDING TO CONDITION CODES
C                 REPTITLE = REPORT TITLE
C-------------------------------------------------------------------------------
```

FILE: HAUL.INC

```
      CHARACTER*16 HREM(101)
      CHARACTER*4 HSMON(101)
      REAL HSSMLAT(101),HSSMLONG(101),HSFMLAT(101),HSFMLONG(101),
     R     HSLEN(101),HFST(101),HLST(101),HSDUR(101),HSVEL(101)
      INTEGER HGT(101),HSSD(101),HSST(101),HSFD(101),HSFT(101),
     I        HSSDLAT(101),HSSDLONG(101),HSFDLAT(101),
     N        HSFDLONG(101),HSDIR(101),HNG(101),HTSC(101),HREG(101),
     T        HAREA(101),HREMN(101),HSDEP1(101),HSDEP2(101),
     E        HDAY(101),HYEAR(101),HSETNUM(101),HLOC(101),
     G        HSDEP3(101),HSDEP4(101)
      COMMON/HAUL/HSSMLAT,HSSMLONG,HSFMLAT,HSFMLONG,HSLEN,HSETNUM,
     C            HFST,HLST,HTSC,HGT,HSSD,HSST,HSFD,HSFT,HSSDLAT,
     O            HSSDLONG,HSFDLAT,HSFDLONG,HSDIR,HNG,HSDEP1,HSDEP2,
     M            HREG,HAREA,HREM,HREMN,HSDUR,HYEAR,HSMON,HDAY,HLOC,
     M            HSVEL,HSDEP3,HSDEP4
C-------------------------------------------------------------------------------
C   HSETNUM  = SET NUMBER                              (10:12)
C   HGT      = GEAR TYPE                               (19:21)
C   HREG     = REGION CODE OF SET                      (13:14)
C   HAREA    = AREA CODE OF SET                        (15:16)
C   HLOC     = LOCALITY CODE OF SET                    (17:18)
C   HSSD     = SET START DATE                          ( 4: 9)
C   HYEAR    = YEAR OF HAUL
C   HSMON    = MONTH OF HAUL
C   HDAY     = DAY OF HAUL
C   HSST     = SET START TIME                          (19:22)
C   HSFD     = SET FINISH DATE
C   HSFT     = SET FINISH TIME
C   HSDUR    = DURATION OF SET                         (50:53)
C   HSDEP1   = DEPTH 1                                 (66:69)
C   HSDEP2   = DEPTH 2                                 (70:73)
C   HSDEP3   = DEPTH 3                                 (62:65)
C   HSDEP4   = DEPTH 4                                 (74:77)
C   HSSDLAT  = SET START DEGREES LATITUDE              (26:27)
C   HSSMLAT  = SET START MINUTES LATITUDE              (28:30)
C   HSSDLONG = SET START DEGREES LONGDITUDE            (31:32)
C   HSSMLONG = SET START MINUTES LONGDITUDE            (33:35)
C   HSFDLAT  = SET FINISH DEGREES LATITUDE             (36:37)
C   HSFMLAT  = SET FINISH MINUTES LATITUDE             (38:40)
C   HSFDLONG = SET FINISH DEGREES LONGDITUDE           (41:42)
C   HSFMLONG = SET FINISH MINUTES LONGDITUDE           (43:45)
C   HSDIR    = SET DIRECTION (DEG TRUE)                (54:56)
C   HSLEN    = SET LENGTH                              (57:59)
C   HNG      = NUMBER OF GROUPS
C   HREM     = REMARK
C   HREMN    = REMARK CODE NUMBER
C   HFST     = FIRST SURFACE TEMPERATURE
C   HLST     = LAST SURFACE TEMP
C   HTSC     = TOTAL TARGET CATCH  (WEIGHT)
```

```
FILE: PROCIN.FOR

C234567
      SUBROUTINE JUL(IDATE,YEAR,MONTH,DAY)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE CONVERTS AN INTEGER REPRESENTATION OF DATE TO
C   YEAR MONTH AND DAY
C
C IMPORTS : IDATE
C EXPORTS : YEAR,MONTH,DAY
C-------------------------------------------------------------------------------
         INTEGER IDATE,YEAR,DAY,TEMP
         CHARACTER*4 LMONTH(12)/'JAN.','FEB.','MAR.','APR.','MAY ',
     C        'JUNE', 'JULY','AUG.','SEPT','OCT.','NOV.','DEC.'/,MONTH

         TEMP = IDATE
         YEAR = TEMP/10000
         TEMP = MOD(TEMP,10000)
         MONTH = LMONTH(TEMP/100)
         DAY = MOD(TEMP,100)
      RETURN
      END
      SUBROUTINE PROCDATA
C-------------------------------------------------------------------------------
C THIS SUBROUTINE PROCESSES THE THAT WAS DATA READ IN
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : CATCH
C                     VARIABLES : CCODE,CNAME,CGROUP,CEST
C                        : HAUL
C                              : HSSD,HSDUR,HSST,HSFD,HSFT,HYEAR,HSMON,HDAY,
C                                HREM,HREMN
C                        : OUTPUT
C                              : REPTITLE,REMARKS
C                        : UNITS
C                              : WTUNIT
C
C PROCEDURES AND FUNCTIONS CALLED :GETNAME,GETEST,SORT,TOTAL,JUL,DIFF
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
$INCLUDE:'HAUL.INC'
         DO 15 I = 1,CNUM,1
C GETTING ASSOCIATED NAME AND GROUP FOR FISH THAT WAS PART OF CATCH
            CALL GETNAME(CCODE(I),CNAME(I),CGROUP(I))
C GETTING ASSOCIATED ESTIMATED AVERAGE WEIGHT FOR FISH THAT WAS PART OF CATCH
            CALL GETEST(CCODE(I),CEST(I))
  15     CONTINUE

         CALL TOTAL
         CALL SORT
         CALL DIFFINIT(HSSD(1)/10000)
         DO 10 I = 1,HNUM,1
C CALCULATING THE DURATION OF THE HAUL
```

```
C             HSDUR(I)=DIFF(HSSD(I),HSST(I),HSFD(I),HSFT(I))
              HSDUR(I)=INT(HSDUR(I)/60)+FLOAT(MOD(INT(HSDUR(I)),60))/100.0
C TRANSFORMING INTEGER REPRESENTATION OF DATE TO YEAR , MONTH , DAY
              CALL JUL(HSSD(I),HYEAR(I),HSMON(I),HDAY(I))
C GETTING ACTUAL REMARK FROM REMARK NUMBER
              HREM(I) = REMARKS(HREMN(I)+1)
   10       CONTINUE
            DO 11 I = 1,3,1
C UPDATING REPORT TITLE TO REFLECT ACTUAL WEIGHT UNITS
              WRITE(REPTITLE(I)(11:13),FMT='(A3)') WTUNIT
   11       CONTINUE
        RETURN
        END
        SUBROUTINE DIFFINIT(YEAR)
C-------------------------------------------------------------------------------
C IMPORTS : YEAR
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : ZZZZ
C                     VARIABLES : SUMDAYS
C-------------------------------------------------------------------------------
            INTEGER YEAR
            INTEGER SUMDAYS(13)
            COMMON/ZZZZ/SUMDAYS
            SUMDAYS(1)=0
            SUMDAYS(2)=31
            SUMDAYS(3)=59
            SUMDAYS(4)=90
            SUMDAYS(5)=120
            SUMDAYS(6)=151
            SUMDAYS(7)=181
            SUMDAYS(8)=212
            SUMDAYS(9)=243
            SUMDAYS(10)=273
            SUMDAYS(11)=304
            SUMDAYS(12)=334
            SUMDAYS(13)=365

            IF (MOD(YEAR,4).EQ.0) THEN
               DO 10 I = 2,13,1
                  SUMDAYS(I) = SUMDAYS(I) + 1
    10         CONTINUE
            ENDIF
        RETURN
        END


        REAL FUNCTION DIFF(SD,ST,FD,FT)
C-------------------------------------------------------------------------------
C IMPORTS : SD,ST,FD,FT
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : ZZZZ
C                     VARIABLES : SUMDAYS
C-------------------------------------------------------------------------------
            INTEGER SD,ST,FD,FT,SUMDAYS(13),SMONTH,FMONTH,SHOUR,FHOUR,
```

```
     I            SMIN,FMIN,SDAY,FDAY,TDIFF,FNUMH,SNUMH
          COMMON/ZZZZ/SUMDAYS

          SMONTH = MOD(SD,10000)/100
          FMONTH = MOD(FD,10000)/100
          SHOUR = ST/100
          SDAY = MOD(SD,100)
          FHOUR = FT/100
          FDAY = MOD(FD,100)
          SMIN = MOD(ST,100)
          FMIN = MOD(FT,100)
          FNUMH = (SUMDAYS(FMONTH)+FDAY)*24 + FHOUR
          SNUMH = (SUMDAYS(SMONTH)+SDAY)*24 + SHOUR
          TDIFF = FNUMH - SNUMH
          IF (SMIN.GT.FMIN) THEN
             DIFF = TDIFF-1+(FMIN+60-SMIN)/100.0
          ELSE
             DIFF = TDIFF+(FMIN-SMIN)/100.0
          ENDIF
        RETURN
        END
        SUBROUTINE ESTIMATE
C-------------------------------------------------------------------------------
C THIS SUBROUTINE ESTIMATES WEIGHTS OR PIECES THAT WERE NOT SUPPLIED BY USING
C THE ESTIMATED WEIGHT/PIECE SUPPLIED BY THE USER
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : CNUM,HNUM
C                        : WPESTIMATE
C                              : WPEST
C                        : CATCH
C                              : CPIECE,CWEIGHT,CEST
C
C PROCEDURES CALLED : TOTAL,SORT
C-------------------------------------------------------------------------------
        INTEGER C,H
$INCLUDE:'COMMON.INC'
        C = CNUM + 1
        H = HNUM + 1
        DO 30 I=1,CNUM,1
           WPEST(I,H,1)=' '
           WPEST(I,H,2)=' '
   30   CONTINUE
        DO 31 J=1,HNUM,1
           WPEST(C,J,1)=' '
           WPEST(C,J,2)=' '
   31   CONTINUE
        DO 25 I=1,CNUM,1
           DO 26 J = 1,HNUM,1
              IF (WPEST(I,J,2).EQ.'*') THEN
C ESTIMATE WEIGHT
                 CWEIGHT(I,J) = CPIECE(I,J)*CEST(I)
```

```
                  WPEST(I,H,2) = '*'
                  WPEST(C,J,2) = '*'
               ELSE
                  WPEST(I,J,2) = ' '
               ENDIF
               IF (WPEST(I,J,1).EQ.'*') THEN
C ESTIMATE PIECES
                  CPIECE(I,J) = CWEIGHT(I,J)/CEST(I)
                  WPEST(I,H,1) = '*'
                  WPEST(C,J,1) = '*'
               ELSE
                  WPEST(I,J,1) = ' '
               ENDIF
   26       CONTINUE
   25    CONTINUE
         CALL TOTAL
         CALL SORT
      RETURN
      END

      SUBROUTINE TOTAL
C-------------------------------------------------------------------------------
C THIS SUBROUTINE TOTAL THE COLUMNS AND ROWS OF 'CPIECE' AND 'CWEIGHT' STORING
C THE TOTALS AT THE END OF EACH COLUMN AND ROW. THE GRAND TOTAL FOR EACH IS
C STORED IN THE BOTTOM RIGHT CORNER (CNUM+1,HNUM+1).
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : CNUM,HNUM
C                        : CATCH
C                                 : CWEIGHT,CPIECE
C-------------------------------------------------------------------------------

      INTEGER C,H
$INCLUDE:'COMMON.INC'
      C = CNUM + 1
      H = HNUM + 1
      CWEIGHT(C,H) = 0
      CPIECE(C,H) = 0
      DO 20 I = 1,HNUM,1
         CPIECE(C,I)=0
         CWEIGHT(C,I)=0
20    CONTINUE
      DO 30 I=1,CNUM,1
         CWEIGHT(I,H)=0.0
         CPIECE(I,H)=0
         DO 40 J=1,HNUM,1
            IF (CWEIGHT(I,J).GT.0) THEN
               CWEIGHT(I,H) = CWEIGHT(I,H) + CWEIGHT(I,J)
               CWEIGHT(C,J) = CWEIGHT(C,J) + CWEIGHT(I,J)
            ENDIF
            IF (CPIECE(I,J).GT.0) THEN
               CPIECE(C,J) = CPIECE(C,J) + CPIECE(I,J)
```

```
                  CPIECE(I,H) = CPIECE(I,H) + CPIECE(I,J)
               ENDIF
 40         CONTINUE
            CPIECE(C,H) = CPIECE(C,H) + CPIECE(I,H)
            CWEIGHT(C,H) = CWEIGHT(C,H) + CWEIGHT(I,H)
 30      CONTINUE
      RETURN
      END

      SUBROUTINE SORT
C-------------------------------------------------------------------------------
C THIS SUBROUTINE SORTS THE CATCH DATA ON A PRIMARY KEY (GROUP NUMBER) AND
C A SECONDARY KEY (WEIGHT OR # OF PIECES ACCORDING TO 'CWORP'). A SELECTION
C SORT IS USED.
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : CNUM , HNUM
C                        : CATCH
C                              : CGROUP,CPIECE,CWEIGHT,CWORP
C
C PROCEDURES CALLED : SWITCH
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
          INTEGER LIM1,LIM2,POS1,POS2,MINPOS,H
          H = HNUM + 1
          LIM1 = CNUM-1
          DO 10 POS1=1,LIM1,1
             MINPOS= POS1
             LIM2 = POS1+1
             DO 20 POS2 = LIM2,CNUM,1
                IF (CGROUP(POS2).LT.CGROUP(MINPOS)) THEN
                   MINPOS = POS2
                ELSEIF (CGROUP(POS2).EQ.CGROUP(MINPOS)) THEN
                   IF (CWORP.EQ.'W') THEN
                      IF(CWEIGHT(POS2,H).GT.CWEIGHT(MINPOS,H))
     I                   MINPOS = POS2
                   ELSE
                      IF(CPIECE(POS2,H).GT.CPIECE(MINPOS,H))
     I                   MINPOS = POS2
                   ENDIF
                ENDIF
 20          CONTINUE
             IF (MINPOS.NE.POS1) CALL SWITCH(POS1,MINPOS)
 10       CONTINUE

      RETURN
      END

      SUBROUTINE SWITCH(P1,P2)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE SWITCHES TWO FISH IN THE CATCH LIST.
C
```

```
C IMPORTS : P1,P2
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : HNUM
C                        : CATCH
C                              : CCODE,CGROUP,CNAME,CEST,CPIECE,CWEIGHT
C                        : WPESTIMATE
C                              : WPEST
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
          CHARACTER*1 T5
          CHARACTER*3 T1
          INTEGER T2,P1,P2,I
          CHARACTER*16 T3
          REAL T4

          T1 = CCODE(P1)
          CCODE(P1)=CCODE(P2)
          CCODE(P2)=T1

          T2=CGROUP(P1)
          CGROUP(P1)=CGROUP(P2)
          CGROUP(P2)=T2

          T3=CNAME(P1)
          CNAME(P1)=CNAME(P2)
          CNAME(P2)=T3

          T4 = CEST(P1)
          CEST(P1) = CEST(P2)
          CEST(P2)=T4

          DO 20 I=1,HNUM+1,1
             T2 = CPIECE(P1,I)
             CPIECE(P1,I) = CPIECE(P2,I)
             CPIECE(P2,I) = T2

             T4 = CWEIGHT(P1,I)
             CWEIGHT(P1,I) = CWEIGHT(P2,I)
             CWEIGHT(P2,I) = T4

             T5 = WPEST(P1,I,1)
             WPEST(P1,I,1) = WPEST(P2,I,1)
             WPEST(P2,I,1) = T5

             T5 = WPEST(P1,I,2)
             WPEST(P1,I,2) = WPEST(P2,I,2)
             WPEST(P2,I,2) = T5

  20      CONTINUE

      RETURN
      END
```

FILE: QUALIFY.FOR

```
C234567
      SUBROUTINE QUALIFY
C-------------------------------------------------------------------------------
C THIS SUBROUTINE DETERMINES WHICH FISH IN THE CATCH WILL QUALIFY TO PRINT.
C IF THERE ARE MORE SPECIES IN A GROUP THAN THERE ARE ALLOWED THE SPECIES
C WITH THE LEAST WEIGHT OR NUMBER OF PIECES (ACCORDING TO 'CWORP') WILL NOT
C BE PRINTED.
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : CNUM
C                        : CATCH
C                              : CGOTHERS,CGROUP
C                        : GROUP
C                              : GMAX
C-------------------------------------------------------------------------------
        INTEGER GOLD,GOLDNUM
$INCLUDE:'COMMON.INC'
        GOLD = 0
        GOLDNUM = 0
        DO 10 I = 1 , CNUM
           IF (GOLD.NE.CGROUP(I)) THEN
              GOLDNUM = 1
              GOLD = CGROUP(I)
           ELSE
              GOLDNUM = GOLDNUM + 1
           ENDIF
           CGOTHERS(I) = (GOLDNUM.GT.GMAX(CGROUP(I)))
  10    CONTINUE

      RETURN
      END

      SUBROUTINE PRQUALIFY
C-------------------------------------------------------------------------------
C THIS SUBROUTINE WRITES TO THE OUTPUT FILE THE LIST OF FISH THAT QUALIFY TO
C PRINT.
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : HNUM
C                        : CATCH
C                              : CGOTHERS,CGROUP,CCODE,CNAME
C                        : GROUP
C                              : GMAX,GNAME
C                        : OUTPUT
C                              : OUTF
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
```

```
      WRITE(OUTF,100) CHAR(12)
100   FORMAT(A1/,' SPECIES TO QUALIFY TO PRINT ')

      WRITE(OUTF,110) (I,I=1,10)
      WRITE(OUTF,120) (GMAX(I),I=1,10)
110   FORMAT(/15X,'GROUP',10(1X,I5))
120   FORMAT(' MAXIMUM SPECIES/GRP',10(1X,I5))

      WRITE(OUTF,125)
125   FORMAT(//' GRP',2X,'CODE',2X,'NAME')
      DO 10 I = 1,CNUM,1
         IF (.NOT. CGOTHERS(I))
     I      WRITE(OUTF,130)CGROUP(I),CCODE(I),CNAME(I)
 10   CONTINUE
130   FORMAT(1X,I3,2X,A3,3X,A16)
      RETURN
      END

      SUBROUTINE SUMMARY
C-------------------------------------------------------------------------------
C THIS SUBROUTINE WRITES TO THE OUTPUT FILE A SUMMARY OF THE DATA PROVIDED.
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : HNUM,CNUM
C                        : CATCH
C                              : CGROUP,CCODE,CNAME,CWEIGHT,CPIECE,CEST
C                        : UNITS
C                              : WTUNIT
C                        : OUTPUT
C                              : OUTF
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
      INTEGER I,H,C
      H = HNUM + 1
      C = CNUM + 1
      WRITE(OUTF,9) CHAR(12),HNUM
 9    FORMAT(A1/,80('-'),/' TOTAL NUMBER OF HAULS FOR CRUISE = ',I3)
      WRITE(OUTF,8)
 8    FORMAT(' TOTAL CATCH IN WEIGHTS AND PIECES, ESTIMATED ',
     F            'AVERAGE WEIGHT.')
      WRITE(OUTF,10)  WTUNIT
10    FORMAT(' GRP',2X,'CODE',3X,'SPECIES',15X,A3,4X,'PIECES'
     F      ,5X,'ESTIMATED',3X,'CALCULATED',/53X,'AVE WT.',6X,
     0       'AVE WT.'/80(' '))
      DO 60 I=1,CNUM,1
         RATIO = 0
         IF (CPIECE(I,H).NE.0) RATIO=CWEIGHT(I,H)/CPIECE(I,H)

         WRITE(OUTF,110) CGROUP(I),CCODE(I),CNAME(I),CWEIGHT(I,H),
     W            CPIECE(I,H),CEST(I),RATIO
60    CONTINUE
110   FORMAT(1X,I3,3X,A3,3X,A16,2X,F8.0,2X,I6,5X,F9.2,3X,F10.2)
```

```
      WRITE(OUTF,120) WTUNIT,CWEIGHT(C,H),CPIECE(C,H),CNUM
120   FORMAT(80('~')/,' TOTAL ',A3,'= ',F6.0,' TOTAL PIECES = ',I6,
     F          ' NUMBER OF SPECIES PRESENT = ',I4,/80('~'))
      RETURN
      END
      SUBROUTINE ESUMMARY
C-------------------------------------------------------------------------------
C THIS SUBROUTINE WRITES TO THE OUTPUT FILE A SUMMARY OF THE DATA PROVIDED AND
C ALSO IF SOME DATA WAS NOT SUPPLIED AN ESTIMATE IS PROVIDED.
C
C IMPORTS : NONE
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : HNUM,CNUM
C                        : CATCH
C                              : CGROUP,CCODE,CNAME,CWEIGHT,CPIECE,CEST
C                        : UNITS
C                              : WTUNIT
C                        : OUTPUT
C                              : OUTF
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
      INTEGER I,PIECE,C,H
      REAL WEIGHT,AVEWT,PPCT,WPCT
      CHARACTER*1 WEST,PEST

      C = CNUM + 1
      H = HNUM + 1
      WRITE(OUTF,9) CHAR(12),HNUM
 9    FORMAT(A1/,80('~')/,' TOTAL NUMBER OF HAULS FOR CRUISE = ',I3)
      WRITE(OUTF,8)
 8    FORMAT(' TOTAL CATCH IN WEIGHTS AND PIECES, ESTIMATED ',
     F           'AVERAGE WEIGHT.')
      WRITE(OUTF,10)  WTUNIT
10    FORMAT(T74,'AVERAGE',/' GRP',2X,'CODE',3X,'SPECIES',
     F     15X,A3,4X,'PERCENT',4X,'PIECES',4X,'PERCENT','  WT/PIECE',
     0     /80('~'))
      DO 60 I=1,CNUM,1

         WEIGHT = CWEIGHT(I,H)
         PIECE = CPIECE(I,H)
         AVEWT = 0.0
         IF (PIECE.NE.0) AVEWT = WEIGHT/PIECE
         WEST = WPEST(I,H,2)
         PEST = WPEST(I,H,1)
         WPCT = 0
         IF (CWEIGHT(C,H).NE.0) WPCT = WEIGHT/CWEIGHT(C,H) *100.0
         PPCT = 0
         IF (CPIECE(C,H).NE.0)
     I       PPCT = FLOAT(PIECE)/FLOAT(CPIECE(C,H))*100.0
         WRITE(OUTF,110) CGROUP(I),CCODE(I),CNAME(I),WEIGHT,WEST,
     W                   WPCT,PIECE,PEST,PPCT,AVEWT
60    CONTINUE
110   FORMAT(1X,I3,3X,A3,3X,A16,2X,F8.0,A1,3X,F6.2,4X,I6,A1,3X,
```

```
     F           F6.2,3X,F8.2)
         WRITE(OUTF,120) WTUNIT,CWEIGHT(C,H),CPIECE(C,H),CNUM
120      FORMAT(80('~')/,' TOTAL ',A3,'= ',F6.0,' TOTAL PIECES = ',I6,
     F           ' NUMBER OF SPECIES PRESENT = ',I4,/80('~'))
      RETURN
      END
```

FILE: FINDFISH.FOR

```
C234567
      SUBROUTINE GETNAME(CODE,NAME,GROUP)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE DETERMINES THE NAME AND GROUP OF A GIVEN FISH CODE
C IF THE CODE OF THE FISH IS NOT AMONG THE KNOWN FISH THEN
C    A : GROUP IS RETURNED AS 11
C    B : NAME IS RETURNED AS 'FISH NOT IN LIST'
C
C IMPORTS : CODE
C EXPORTS : NAME,GROUP
C COMMON BLOCK ACCESSED : FISHNAMES
C              VARIABLES : FCODE,FNAME,FGROUP
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
         CHARACTER*16 NAME
         CHARACTER*3 CODE
         INTEGER GROUP,POS

         DO 10 POS = 1, FNUM
            IF (CODE.EQ.FCODE(POS)) GOTO 20
 10      CONTINUE
 20      IF (CODE.EQ.FCODE(POS)) THEN
            NAME = FNAME(POS)
            GROUP = FGROUP(POS)
         ELSE
            NAME = 'FISH NOT IN LIST'
            GROUP = 11
         ENDIF
      RETURN
      END
      SUBROUTINE GETEST(CODE,EST)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE DETERMINES THE ESTIMATED AVERAGE FOR A GIVEN FISH CODE
C IF THE CODE OF THE FISH IS NOT AMONG THE KNOWN FISH THEN
C    A : EST IS RETURNED AS 1.0
C
C IMPORTS : CODE
C EXPORTS : EST
C COMMON BLOCK ACCESSED : AVEWEIGHT
C              VARIABLES : ACODE,AWEIGHT
C-------------------------------------------------------------------------------

$INCLUDE:'COMMON.INC'
         CHARACTER*3 CODE
         REAL EST
         INTEGER POS

         DO 10 POS = 1,ANUM,1
            IF (CODE.EQ.ACODE(POS)) GOTO 20
 10      CONTINUE
 20      IF (ACODE(POS).EQ.CODE) THEN
            EST = AWEIGHT(POS)
```

FILE: FILEIO.FOR

```
C234567
      SUBROUTINE FISHINIT(FILENAME)
C-------------------------------------------------------------------------
C THIS SUBROUTINE INITIALIZES THE LIST OF KNOWN FISH NAMES,CODES AND
C   THEIR CORRESPONDING GROUPS
C
C IMPORTS : FILENAME
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : FISH
C                      VARIABLES : FGROUP,FCODE,FNAME
C                       : LISTSIZES
C                                 : FNUM
C-------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
        CHARACTER*80 FILENAME


        OPEN(UNIT=10,FILE=FILENAME,STATUS='OLD',ERR=111)

        DO 30 FNUM = 1,300,1
           READ(10,20,END=40)FGROUP(FNUM),FCODE(FNUM),FNAME(FNUM)
 30     CONTINUE
 40     FNUM = FNUM-1
 20     FORMAT(I5,2X,A3,4X,A16)
        CLOSE(10)
      RETURN
 111  STOP ' ERROR OPENING FISH NAME FILE'
      END

      SUBROUTINE LINEINIT(FILENAME)
C-------------------------------------------------------------------------
C THIS SUBROUTINE INITIALIZES THE LIST OF LINE TITLES
C
C IMPORTS : FILENAME
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : OUTPUT
C                      VARIABLES : LINETITLE
C                       : LISTSIZES
C                                 : LNUM
C-------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
        CHARACTER*80 FILENAME


        OPEN(UNIT=10,FILE=FILENAME,STATUS='OLD',ERR=111)

        DO 30 LNUM = 1,31,1
           READ(10,20,END=40)LINETITLE(LNUM)
 30     CONTINUE
 40     LNUM = LNUM - 1
 20     FORMAT(A20)
        CLOSE(10)
```

```
      RETURN
 111  STOP 'ERROR OPENING LINE TITLE FILE'
      END

      SUBROUTINE REMINIT(FILENAME)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE INITIALIZES THE LIST OF REMARKS
C
C IMPORTS : FILENAME
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : OUTPUT
C                       VARIABLES : REMARKS
C                        : LISTSIZES
C                                 : REMNUM
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
         CHARACTER*80 FILENAME


         OPEN(UNIT=10,FILE=FILENAME,STATUS='OLD',ERR=111)

         DO 30 REMNUM = 1,31,1
            READ(10,20,END=40)REMARKS(REMNUM)
 30      CONTINUE
 40      REMNUM = REMNUM - 1
 20      FORMAT(A20)
         CLOSE(10)
      RETURN
 111  STOP 'ERROR OPENING REMARKS FILE'
      END

      SUBROUTINE AVEINIT(FILENAME)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE INITIALIZES THE LIST OF KNOWN FISH AVERAGE WTS
C
C IMPORTS : FILENAME
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : AVEWEIGHT
C                       VARIABLES : ACODE,AWEIGHT
C                        : LISTSIZES
C                                 : ANUM
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
         CHARACTER*80 FILENAME

         OPEN(UNIT=10,FILE=FILENAME,STATUS='OLD',ERR=111)
         DO 30 ANUM=1,300,1
            READ(10,25,END=40) ACODE(ANUM),AWEIGHT(ANUM)
   30    CONTINUE

   40    ANUM=ANUM-1
   25    FORMAT(A3,F8.2)
         CLOSE(10)
      RETURN
```

```
 111  STOP 'ERROR OPENING FISH AVERAGE FILE'
      END

      SUBROUTINE DATAINIT(FILENAME)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE READS IN THE DATA FOR THE HAUL
C
C IMPORTS : FILENAME
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : HAUL
C                     VARIABLES : HSSMLAT,HSSMLONG,HSFMLAT,HSFMLONG,HSLEN,
C                                 HSETNUM,,HFST,HLST,HTSC,HGT,HSSD,HSST,HSFD,
C                                 HSFT,HSSDLAT,HSSDLONG,HSFDLONG,HSDIR,HNG,
C                                 HSDEP1,HSDEP2,HREG,HAREA,HREMN
C                      : LISTSIZES
C                              : CNUM,HNUM
C                      : CATCH
C                              : CPIECE,CWEIGHT,CCODE,
C                      : WPESTIMATE
C                              : WPEST
C                      : UNITS
C                              : WTUNIT,TUNITS,DUNITS
C-------------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
$INCLUDE:'HAUL.INC'

      CHARACTER*80 FILENAME*80,LINE*80,CODE*3,EST*1
      INTEGER I,CATCHP,PIN,CTYPE
      REAL CATCHW

      OPEN(UNIT=10,FILE=FILENAME,STATUS='OLD',ERR=111)
      HNUM = 0

30    READ(10,20,END=40) LINE
         IF (LINE(1:1).NE.'C') THEN
            READ(LINE(79:80),FMT='(I2)')CTYPE
            IF (CTYPE.EQ.3) THEN
               WTUNIT = ' KG'
               READ(LINE(19:23),FMT='(I5)') HTSC(HNUM)
               READ(LINE(26:37),1)CODE,CATCHP,CATCHW
1              FORMAT(A3,I4,F5.0)
               DO 25 PIN = 1,CNUM,1
                  IF (CODE.EQ.CCODE(PIN)) GOTO 15
25             CONTINUE
15             IF (PIN.GT.CNUM) THEN
                  CNUM = CNUM+1
                  CCODE(CNUM) = CODE
               ENDIF
               IF (CATCHW.GT.0 .AND. CATCHP.LE.0)
     I            WPEST(PIN,HNUM,1)='*'
               IF (CATCHP.GT.0 .AND. CATCHW.LE.0)
     I            WPEST(PIN,HNUM,2)='*'
               CPIECE(PIN,HNUM) = CATCHP
               CWEIGHT(PIN,HNUM) = CATCHW
```

```
                ELSEIF (CTYPE.EQ.1) THEN
                   HNUM = HNUM + 1
                   READ(LINE(4:77),27)HSSD(HNUM),HSETNUM(HNUM),
      R                 HREG(HNUM),HAREA(HNUM),HLOC(HNUM),HGT(HNUM),
      E                 HSSDLAT(HNUM),HSSMLAT(HNUM),HSSDLONG(HNUM),
      A                 HSSMLONG(HNUM),HSFDLAT(HNUM),HSFMLAT(HNUM),
      D                 HSFDLONG(HNUM),HSFMLONG(HNUM),HSST(HNUM),
      L                 HSDUR(HNUM),HSDIR(HNUM),HSLEN(HNUM),HSVEL(HNUM),
      I                 HSDEP3(HNUM),HSDEP1(HNUM),HSDEP2(HNUM),
      N                 HSDEP4(HNUM)

                   HSVEL(HNUM) = HSVEL(HNUM)/10.0
C                   READ(LINE(4:6),FMT='(I3)')HSETNUM(HNUM)
C                   READ(LINE(7:78),26) HGT(HNUM),HREG(HNUM),HAREA(HNUM),
C     R                 HSSD(HNUM),HSST(HNUM),HSFD(HNUM),HSFT(HNUM),
C     E                 HSDEP1(HNUM),HSDEP2(HNUM),HSSDLAT(HNUM),
C     A                 HSSMLAT(HNUM),HSSDLONG(HNUM),HSSMLONG(HNUM),
C     D                 HSFDLAT(HNUM),HSFMLAT(HNUM),HSFDLONG(HNUM),
C     L                 HSFMLONG(HNUM),HSDIR(HNUM),HSLEN(HNUM),HNG(HNUM),
C     I                 HREMN(HNUM),HFST(HNUM),HLST(HNUM),HTSC(HNUM)

                   HSSMLAT(HNUM) = HSSMLAT(HNUM)/10.0
                   HSSDLONG(HNUM) = HSSDLONG(HNUM) + 100
                   HSSMLONG(HNUM) = HSSMLONG(HNUM)/10.0
                   HSFMLAT(HNUM) = HSFMLAT(HNUM)/10.0
                   HSFDLONG(HNUM) = HSFDLONG(HNUM) + 100
                   HSFMLONG(HNUM) = HSFMLONG(HNUM)/10.0
                   HSLEN(HNUM) = HSLEN(HNUM)/10.0
                   HFST(HNUM) = HFST(HNUM)/10.0
                   HLST(HNUM) = HLST(HNUM)/10.0
                ELSEIF (CTYPE.NE.2) THEN
                   STOP '-- INVALID CARD TYPE -- DATAINIT'
                ENDIF
             ENDIF
          GOTO 30
   40     CLOSE(10)
   20     FORMAT(A80)
   26     FORMAT(I3,I1,I2,I6,3I4,2I3,4(I2,F3.0),I3,F3.0,I3,I2,2F3.0,I5)
   27     FORMAT(I6,I3,3I2,I3,4X,4(I2,F3.0),I4,F4.0,I3,F3.0,F2.0,4I4)
       RETURN
  111  STOP 'ERROR OPENING DATA FILE'
       END
       SUBROUTINE GROUPINIT(FILENAME)
C-------------------------------------------------------------------------
C THIS SUBROUTINE INITIALIZES THE LIST OF KNOWN GROUP NAMES
C
C IMPORTS : FILENAME
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : GROUP
C                       VARIABLES : GNAME
C                        : LISTSIZES
C                                  : GNUM
C-------------------------------------------------------------------------
$INCLUDE:'COMMON.INC'
```

```
      CHARACTER*80 FILENAME

      OPEN(UNIT=10,FILE=FILENAME,STATUS='OLD',ERR=111)

      DO 10 GNUM=1,11,1
         READ(10,20,END=30) GNAME(GNUM)
10    CONTINUE
20    FORMAT(A16)
30    GNUM = GNUM-1
      GNAME(11) = 'UNKNOWN GROUP'
      CLOSE(10)
    RETURN
111 STOP 'ERROR OPENING GROUP NAMES FILE'
    END
```

FILE: REPORT.FOR

```
C234567
      SUBROUTINE CLR(PAGE,SIZE)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE CLEARS A 'PAGE' OF LENGTH 'SIZE'.
C
C IMPORTS : SIZE
C EXPORTS : PAGE
C-------------------------------------------------------------------------------
         INTEGER SIZE
         CHARACTER*80 PAGE(SIZE)

         DO 10 I = 1,SIZE,1
            PAGE(I)='                                                                '
  10     CONTINUE
      RETURN
      END
      INTEGER FUNCTION LENSTR(STR)
C-------------------------------------------------------------------------------
C THIS FUNCTION DETERMINES THE LENGTH OF 'STR' TO THE NEAREST 5 CHARACTERS
C
C IMPORTS : STR
C-------------------------------------------------------------------------------
         CHARACTER STR*80
         ILEN = LEN(STR)
         DO 10 I = ILEN-4,1,-5
            IPOS = INDEX(STR(I:ILEN),'     ')
            IF (IPOS.NE.1) GOTO 11
  10     CONTINUE
  11     LENSTR = I+5
      RETURN
      END
      SUBROUTINE PRINT(OUTPUT,PAGE,SIZE)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE WRITES A 'PAGE' OF LENGTH 'SIZE' TO 'OUTF'
C
C IMPORTS : PAGE,LENGTH,SIZE
C EXPORTS : NONE
C
C-------------------------------------------------------------------------------
         INTEGER OUTPUT,SIZE
         CHARACTER PAGE(SIZE)*80,FM*30/'(A74)'/

         WRITE(OUTPUT,10) CHAR(12)
  10     FORMAT(A1)

         DO 15 I=1,SIZE,1
            WRITE(FM(3:4),FMT='(I2.2)') LENSTR(PAGE(I))
            WRITE(OUTPUT,FMT=FM) PAGE(I)
  15     CONTINUE

      RETURN
      END
```

```
      SUBROUTINE REPORT(RNUM,HNUM)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE CALLS THE PROCEDURES TO PRODUCE THE TWO PAGES FOR EACH SET
C OF SIX HAULS.
C
C IMPORTS : RNUM,HNUM
C EXPORTS : NONE
C
C PROCEDURES CALLED : PAGEA,PAGEB
C-------------------------------------------------------------------------------
         INTEGER RNUM,HNUM,LLIM,ULIM,PNUM
         PNUM=1
         DO 10 I = 1,HNUM-6,6
            LLIM = I
            ULIM = I + 5
            CALL PAGEA(RNUM,LLIM,ULIM,PNUM)
            CALL PAGEB(RNUM,LLIM,ULIM,PNUM)
            PNUM = PNUM + 1
 10      CONTINUE
         CALL PAGEA(RNUM,I,HNUM,PNUM)
         CALL PAGEB(RNUM,I,HNUM,PNUM)

      RETURN
      END
      SUBROUTINE PAGEA(RNUM,LLIM,ULIM,PNUM)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE PRODUCES PAGE '<PNUM>.1' OF A TWO PAGE REPORT FOR EACH GROUP
C OF 6 HAULS.
C
C IMPORTS : RNUM,LLIM,ULIM,PNUM
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : OUTPUT
C                     VARIABLES : PAGE1,OUTF,LINETITLE
C                        : HAUL
C                              : HSST,HSDUR,HSSDLAT,HSSMLAT,HSSDLONG,HSSMLONG,
C                                HSDIR,HSFDLAT,HSFMLAT,HSFDLONG,HSFMLONG,
C                                HSLEN,HSDEP1,HSDEP2,HFST,HLST,HNG,HGT,
C                                HTSC,HREM
C
C PROCEDURES CALLED : CLR,THREE,HEADER,PRINT
C-------------------------------------------------------------------------------
         REAL DEPFAC/1.8288/
         INTEGER RNUM,LLIM,ULIM,COLUMN,ROW,PNUM,R
         CHARACTER*30 FM(8)/'(A20,6(1X,I7,1X))','(A20,6(1X,F7.2,1X))',
     D           '(A20,6(1X,F7.1,1X))','(A20,6(1X,I7.3,1X))',
     A           '(A20,6(1X,I3,''-'',I3,1X))','(A20,6(A9))',
     T           ' ',' '/

$INCLUDE: 'COMMON.INC'
$INCLUDE: 'HAUL.INC'
         CALL CLR(PAGE1,66)
         ROW = 1
         CALL THREE(RNUM,PAGE1,ROW,PNUM,'.1')
         CALL HEADER(PAGE1,LLIM,ULIM,ROW)
```

```
C START TIME
        ROW=ROW+2
        WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(4),(HSST(I),
     W                  I=LLIM,ULIM,1)

C DURATION
        ROW=ROW+2
        WRITE(PAGE1(ROW)(1:),FMT=FM(2))LINETITLE(5),(HSDUR(I),
     W                  I=LLIM,ULIM,1)

C START DEGREES LATITUDE
        ROW=ROW+2
        WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(6),(HSSDLAT(I),
     W                  I=LLIM,ULIM,1)
C START MINUTES LATITUDE
        ROW=ROW+1
        WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(7),(HSSMLAT(I),
     W                  I=LLIM,ULIM,1)

C START DEGREES LONG.
        ROW=ROW+2
        WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(8),(HSSDLONG(I),
     W                  I=LLIM,ULIM,1)
C START MINUTES LONG.
        ROW=ROW+1
        WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(9),(HSSMLONG(I),
     W                  I=LLIM,ULIM,1)

C DIRECTION
        ROW=ROW+2
        WRITE(PAGE1(ROW)(1:),FMT=FM(4))LINETITLE(10),(HSDIR(I),
     W                  I=LLIM,ULIM,1)

C FINISH DEGREES LATITUDE
        ROW = ROW +2
        WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(11),(HSFDLAT(I),
     W                  I=LLIM,ULIM,1)
C FINISH MINUTES LATITUDE
        ROW=ROW+1
        WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(12),(HSFMLAT(I),
     W                  I=LLIM,ULIM,1)

C FINISH DEGREES LONG.
        ROW=ROW+2
        WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(13),(HSFDLONG(I),
     W                  I=LLIM,ULIM,1)
C FINISH MINUTES LONG.
        ROW= ROW +1
        WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(14),(HSFMLONG(I),
     W                  I=LLIM,ULIM,1)

C LENGTH
        ROW = ROW+2
```

```
      WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(15),(HSLEN(I),
     W                I=LLIM,ULIM,1)

C DEPTH 1 & 2
      ROW=ROW+2
      WRITE(LINETITLE(16)(1:13),FMT='(A13)') 'DEPTH START  '
      WRITE(PAGE1(ROW)(1:),FMT=FM(5))LINETITLE(16),(HSDEP1(I),
     W          INT(HSDEP1(I)/DEPFAC),I=LLIM,ULIM,1)
      ROW = ROW + 1
      WRITE(LINETITLE(16)(1:13),FMT='(A13)') '      FINISH '
      WRITE(PAGE1(ROW)(1:),FMT=FM(5))LINETITLE(16),(HSDEP2(I),
     W          INT(HSDEP2(I)/DEPFAC),I=LLIM,ULIM,1)
      ROW = ROW + 1
      WRITE(LINETITLE(16)(1:13),FMT='(A13)') '      INTERV.'
      WRITE(PAGE1(ROW)(1:),FMT=FM(5))LINETITLE(16),
     W          (R(HSDEP1(I)*1.0,HSDEP2(I)*1.0),
     R           R(HSDEP1(I)/DEPFAC,HSDEP2(I)/DEPFAC),
     R           I=LLIM,ULIM,1)

C SEE FIGURE NUMBER
      ROW=ROW+2
      WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(17)

C TEMPERATURE 1
      ROW=ROW+2
      WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(18),(HFST(I),
     W                I=LLIM,ULIM,1)
C TEMPERATURE 2
      ROW=ROW+1
      WRITE(PAGE1(ROW)(1:),FMT=FM(3))LINETITLE(19),(HLST(I),
     W                I=LLIM,ULIM,1)

C GEAR TYPE
      ROW=ROW+2
      WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(20),(HGT(I),
     W                I=LLIM,ULIM,1)

C NUMBER OF GROUPS
      ROW=ROW+2
      WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(21),(HNG(I),
     W                I=LLIM,ULIM,1)

C TOTAL TARGET CATCH
      ROW=ROW+2
      WRITE(PAGE1(ROW)(1:),FMT=FM(1))LINETITLE(22),(HTSC(I),
     W                I=LLIM,ULIM,1)

C REMARKS
      ROW=ROW+2
      WRITE(PAGE1(ROW)(1:),FMT=FM(6))LINETITLE(23),(HREM(I),
     W                I=LLIM,ULIM,1)

C PRINT PAGE 1
      CALL PRINT(OUTF,PAGE1,ROW)
```

```
      RETURN
      END

      INTEGER FUNCTION R(ONE,TWO)
         REAL ONE,TWO

         R = INT((ONE+TWO)/20.0) * 10
      RETURN
      END
      SUBROUTINE THREE(RNUM,PAGE,ROW,PNUM,LETT)
C-------------------------------------------------------------------------
C THIS SUBROUTINE PRODUCES THE FIRST FOUR LINES OF BOTH PAGES OF THE REPORT
C
C IMPORTS : RNUM,ROW,PNUM,LETT
C EXPORTS : PAGE
C COMMON BLOCK ACCESSED : OUTPUT
C                    VARIABLES : PHEADER,REPTITLE
C-------------------------------------------------------------------------
$INCLUDE: 'COMMON.INC'
         CHARACTER*2 LETT
         INTEGER RNUM,ROW,PNUM
         CHARACTER*80 PAGE(66)

C PAGE HEADER LINE
         WRITE(PAGE(1)(1:),20) RNUM,REPTITLE(RNUM),PNUM,LETT
  20     FORMAT('APPENDIX TABLE ',I1,' - ',A,T64,'PAGE ',I4,A2)

C IF PAGE NUMBER EQUALS 1 THEN ADD USER SUPPLIED PAGE HEADER LINES
         IF (PNUM.EQ.1) THEN
            DO 10 ROW=2,4
               PAGE(ROW)=PHEADER(ROW-1)
  10        CONTINUE
         ENDIF
         ROW = 4
      RETURN
      END
      SUBROUTINE HEADER(PAGE,LLIM,ULIM,ROW)
C-------------------------------------------------------------------------
C THIS SUBROUTINE PRODUCES THE FIFTH THROUGH SEVENTH LINES FOR BOTH PAGES OF
C THE REPORT.
C
C IMPORTS : LLIM,ULIM,ROW
C EXPORTS : PAGE
C COMMON BLOCK ACCESSED : HAUL
C                    VARIABLES : HSETNUM,HSMON,HDAY,HREG,HAREA
C                       : OUTPUT
C                             : LINETITLE
C-------------------------------------------------------------------------
         INTEGER LLIM,ULIM,ROW
         CHARACTER*80 PAGE(66)
         CHARACTER*30 FM(3)/'(A20,T21,6(1X,I7,1X))',
     C           '(A20,T21,6(1X,A4,I3,1X))',
     H           '(A20,T21,6(1X,I4,I3,1X))'/
```

```
$INCLUDE: 'COMMON.INC'
$INCLUDE: 'HAUL.INC'

C SET NUMBER
         ROW = ROW+2
         WRITE(PAGE(ROW)(1:),FMT=FM(1))LINETITLE(1),(HSETNUM(I),
     W                  I=LLIM,ULIM,1)

C DATE
         ROW = ROW+2
         WRITE(PAGE(ROW)(1:),FMT=FM(2))LINETITLE(2),(HSMON(I),
     W          HDAY(I),I=LLIM,ULIM,1)

C REGION AND AREA
         ROW = ROW+2
         WRITE(PAGE(ROW)(1:),FMT=FM(3))LINETITLE(3),(HREG(I),
     W          HAREA(I),I=LLIM,ULIM,1)
      RETURN
      END

      SUBROUTINE PAGEB(RNUM,LLIM,ULIM,PNUM)
C-------------------------------------------------------------------------
C THIS SUBROUTINE PRODUCES PAGE '<PNUM>.2' OF THE TWO PAGE REPORT FOR EACH SIX
C HAULS.
C
C IMPORTS : RNUM,LLIM,ULIM,PNUM
C EXPORTS : NONE
C COMMON BLOCK ACCESSED : LISTSIZES
C                     VARIABLES : CNUM
C                        : CATCH
C                              : CGROUP,CGOTHERS,CNAME
C                        : GROUP
C                              : GNAME
C
C PROCEDURES CALLED : THREE,HEADER,HEAD2,LINER,PRINT
C-------------------------------------------------------------------------
$INCLUDE: 'COMMON.INC'

         CHARACTER*1 EST,OTE(6),F1E(6),OTHERS*20/'OTHERS'/
         INTEGER RNUM,LLIM,ULIM,PNUM,ROW,GOLD,NUM
         REAL OTOT(6),F1(6),TOL(6)/1,0.005,0.0005,1,0.005,0.0005/
         CHARACTER FM(6)*40/'(1X,I7  ,A1)','(1X,F7.2,A1)',
     D            '(1X,F7.3,A1)','(1X,I7  ,A1)',
     A            '(1X,F7.2,A1)','(1X,F7.3,A1)'/
         CHARACTER NPKG*20/'PIECES'/

         NUM = ULIM - LLIM + 1

         CALL CLR(PAGE2,132)
         ROW = 1
         CALL THREE(RNUM,PAGE2,ROW,PNUM,'.2')
         CALL HEADER(PAGE2,LLIM,ULIM,ROW)
         CALL HEAD2(RNUM,LLIM,ULIM,ROW)
         ROW=ROW+2
```

```
      GOLD = CGROUP(1)
      DO 11 I = 1,6,1
         OTOT(I)= 0.0
         OTE(I) = ' '
 11   CONTINUE

      WRITE(PAGE2(ROW)(1:),FMT='(A16)') GNAME(GOLD)

      DO 20 I = 1,CNUM,1
         IF (CGROUP(I).NE.GOLD) THEN
C IN A NEW GROUP THEN PRINT OTHERS LINE FOR PREVIOUS GROUP AND RESET TOTALS
C ALSO PRINT NEW GROUP NAME
            ROW=ROW+1
            CALL LINER(PAGE2(ROW),FM(RNUM),TOL(RNUM),2,OTHERS,
     C                 OTOT,OTE,NUM,RNUM)
            DO 200 J = 1,NUM,1
               OTE(J) = ' '
               OTOT(J)=0.0
200         CONTINUE
            GOLD = CGROUP(I)
            ROW = ROW + 2
            WRITE(PAGE2(ROW)(1:),FMT='(A16)') GNAME(GOLD)
         ENDIF
         IF (.NOT. CGOTHERS(I)) THEN
C IF SPECIES QUALIFIED TO PRINT THEN PRINT IT
            ROW = ROW + 1
            DO 201 J = 1,NUM,1
               F1(J) = F(I,J+LLIM-1,RNUM,F1E(J))
201         CONTINUE
            CALL LINER(PAGE2(ROW),FM(RNUM),TOL(RNUM),4,CNAME(I),
     C                 F1,F1E,NUM,RNUM)
            IF (I.EQ.1.AND.RNUM.EQ.1) THEN
C IF ON FIRST SPECIES AND REPORT NUMBER ONE THEN PRINT 'PIECES' LINE
               ROW = ROW + 1
               DO 202 J = 1,NUM,1
                  F1(J) = F(I,J+LLIM-1,RNUM+3,F1E(J))
202            CONTINUE
               CALL LINER(PAGE2(ROW),FM(RNUM+3),TOL(RNUM+3),6,NPKG,
     C                    F1,F1E,NUM,RNUM+3)
            ENDIF
         ELSE
C IF SPECIES DID NOT QUALIFY TO PRINT THEN ADD IT TO OTHERS TOTAL
            DO 30 J = 1,NUM,1
               OTOT(J) = OTOT(J) + F(I,LLIM+J-1,RNUM,EST)
               IF (EST.EQ.'*') OTE(J) = EST
30          CONTINUE
         ENDIF
20    CONTINUE
      ROW=ROW+1
      CALL LINER(PAGE2(ROW),FM(RNUM),TOL(RNUM),2,OTHERS,OTOT,OTE,NUM,
     C           RNUM)
      CALL PRINT(OUTF,PAGE2,ROW)
      RETURN
      END
```

```
      SUBROUTINE LINER(LINE,FORM,TOL,FILL,STR1,LIST1,LIST2,UB,RNUM)
C-----------------------------------------------------------------------
C THIS SUBROUTINE PRODUCES A LINE OF PAGE 2
C
C IMPORTS : FORM,TOL,FILL,STR1,LIST1,LIST2,UB,RNUM
C EXPORTS : LINE
C-----------------------------------------------------------------------
         INTEGER UB,FILL,LENG,IS,IE,RNUM
         REAL LIST1(UB)
         CHARACTER LINE*80,FORM*40,STR1*20,LIST2(6)*1,FM*40/'(  X,A  )'/

         IF (FILL.GT.0.AND.FILL.LE.20) THEN
            FM = '(  X,A  )'
            WRITE (FM(2:3),FMT='(I2)') FILL
            WRITE(FM(7:8),FMT='(I2.2)') 20-FILL
         ELSE
            FM = '(A20)'
         ENDIF

         WRITE(LINE(1:20),FMT=FM) STR1
         DO 10 I = 1,UB,1
            IS = 21 + (I-1)*9
            IE = IS+8
            IF (LIST1(I).GE.TOL) THEN
C IF NUMBER IS GREATER THAN A PRESET TOLERANCE THE PRINT IT
               IF(MOD(RNUM,3).NE.1) THEN
C     IF REPORT NUMBER IS 2,3,5, OR 6 OUTPUT NUMBER AS REAL
                  WRITE(LINE(IS:IE),FMT=FORM) LIST1(I),LIST2(I)
               ELSE
C     ELSEIF REPORT NUMBER IS 1 OR 4 OUTPUT NUMBER AS INTEGER
                  WRITE(LINE(IS:IE),FMT=FORM) INT(LIST1(I)),LIST2(I)
               ENDIF
            ELSEIF (LIST1(I).LT.0) THEN
C ELSEIF NUMBER IS LESS THAN 0 THAT INDICATES A 'TRACE' WAS PRESENT
               WRITE(LINE(IS:IE),FMT='(3X,A6)') 'TRACE '
            ELSE
C ELSEIF NUMBER EQUALS 0 THEN PRINT '..'
               WRITE(LINE(IS:IE),FMT='(6X,A2,A1)') '..',LIST2(I)
            ENDIF
  10     CONTINUE
      RETURN
      END
      REAL FUNCTION F(I,J,RNUM,EST)
C-----------------------------------------------------------------------
C THIS FUNCTION RETURNS THE VALUE OF THE Ith CATCH AND Jth HAUL ACCORDING TO
C REPORT NUMBER.
C
C IMPORTS : I,J,RNUM
C EXPORTS : EST
C COMMON BLOCK ACCESSED : CATCH
C                     VARIABLES : CPIECE,CWEIGHT
C                        : HAUL
C                              : HSLEN,HSDUR
C                        : WPESTIMATE
```

```
C                                : WPEST
C-------------------------------------------------------------------------------
         INTEGER I,J,RNUM
         CHARACTER*1 EST
$INCLUDE: 'COMMON.INC'
$INCLUDE: 'HAUL.INC'
         IF (RNUM.LT.4) THEN
C IF REPORT IS BY WEIGHTS
            EST = WPEST(I,J,2)
         ELSE
C ELSEIF REPORT IS BY PIECES
            EST = WPEST(I,J,1)
         ENDIF

         IF (RNUM.EQ.1) THEN
            F = CWEIGHT(I,J)
         ELSEIF (RNUM.EQ.2) THEN
            F = CWEIGHT(I,J)/HSLEN(J)
         ELSEIF (RNUM.EQ.3) THEN
            F = CWEIGHT(I,J)/(HSLEN(J)*HSDUR(J))
         ELSEIF (RNUM.EQ.4) THEN
            F = FLOAT(CPIECE(I,J))
         ELSEIF (RNUM.EQ.5) THEN
            F = CPIECE(I,J)/HSLEN(J)
         ELSE
            F = CPIECE(I,J)/(HSLEN(J)*HSDUR(J))
         ENDIF
      RETURN
      END
      REAL FUNCTION GETDUR(IHAUL)
C-------------------------------------------------------------------------------
C THIS FUNCTION RETURNS THE DURATION OF THE SET IN DECIMAL HOURS
C i.e. 10.50 IS 10 HRS 30 MINUTES
C
C IMPORTS : IHAUL
C
C COMMON BLOCKS ACCESSED : HAUL
C                       VARIABLE : HSDUR
C-------------------------------------------------------------------------------
$INCLUDE: 'HAUL.INC'

         IHOUR = INT(HSDUR(IHAUL))
         GETDUR = IHOUR + (HSDUR(IHAUL)-IHOUR)/0.6
      RETURN
      END
      REAL FUNCTION GETLEN(IHAUL)
C-------------------------------------------------------------------------------
C THIS FUNCTION RETURNS THE LENGTH OF THE CRUISE.
C
C IMPORTS : IHAUL
C
C COMMON BLOCKS ACCESSED : HAUL
C                       VARIABLE : HSLEN
C-------------------------------------------------------------------------------
```

```
$INCLUDE: 'HAUL.INC'
          GETLEN = HSLEN(IHAUL)
        RETURN
        END
        SUBROUTINE HEAD2(RNUM,LLIM,ULIM,ROW)
C-------------------------------------------------------------------------------
C THIS SUBROUTINE PRODUCES THE SECONDARY HEADER LINES FOR PAGE 2
C
C IMPORTS : RNUM,LLIM,ULIM,ROW
C EXPORTS : NONE
C
C PROCEDURES AND FUNCTIONS CALLED : LINER,F,GETDUR,GETLEN
C-------------------------------------------------------------------------------
$INCLUDE: 'COMMON.INC'
          INTEGER RNUM,LLIM,ULIM,ROW,C
          CHARACTER F1EST(6)*1,FM(2)*30/'(1X,F7.2,A1)',
     D              '(1X,I7,A1)'/
          CHARACTER TCW*20/'TOTAL CATCH WEIGHT'/,DIS*20/'DISTANCE'/
          CHARACTER TCP*20/'TOTAL CATCH PIECES'/
          CHARACTER TC1*20/'TOTAL CPUE'/,TC2*20/'TOTAL CPUE /(KM*HRS)'/
          CHARACTER DISH*20/'DISTANCE*HOURS'/
          REAL F1(6),TOL/0.005/

          C = CNUM + 1
          NUM = ULIM-LLIM+1
          IF(RNUM.GT.0.AND.RNUM.LE.3) THEN
             ROW=ROW+2
             DO 201 I = 1,NUM
                F1(I)=F(C,I+LLIM-1,1,F1EST(I))
201          CONTINUE
             CALL LINER(PAGE2(ROW),FM(2),1.0,0,TCW,F1,F1EST,NUM,1)
             IF (RNUM.EQ.1) THEN
                ROW=ROW+2
                DO 216 I = 1,NUM
                   F1(I)=F(C,I+LLIM-1,4,F1EST(I))
216             CONTINUE
                CALL LINER(PAGE2(ROW),FM(2),1.0,0,TCP,F1,F1EST,NUM,1)
             ELSEIF (RNUM.EQ.2) THEN
                ROW=ROW+1
                DO 202 I = 1,NUM,1
                   F1(I) = GETLEN(I+LLIM-1)
                   F1EST(I)=' '
202             CONTINUE
                CALL LINER(PAGE2(ROW),FM(1),TOL,0,DIS, F1,F1EST,NUM,2)

                ROW = ROW + 2
                DO 203 I = 1,NUM,1
                   F1(I)=F(C,I+LLIM-1,RNUM,F1EST(I))
203             CONTINUE
                CALL LINER(PAGE2(ROW),FM(1),TOL,0,TC1,F1,F1EST,NUM,2)
             ELSEIF (RNUM.EQ.3) THEN
                ROW=ROW+1
                DO 204 I = 1,NUM,1
                   F1(I)=GETDUR(I+LLIM-1)*GETLEN(I+LLIM-1)
```

```
               F1EST(I) = ' '
204            CONTINUE
               CALL LINER(PAGE2(ROW),FM(1),TOL,0,DISH,F1,F1EST,NUM,2)
               ROW=ROW+2
               DO 205 I = 1,NUM,1
                  F1(I)=F(C,I+LLIM-1,RNUM,F1EST(I))
205            CONTINUE
               CALL LINER(PAGE2(ROW),FM(1),TOL,0,TC2,F1,F1EST,NUM,2)
            ENDIF
         ELSEIF (RNUM.GT.3) THEN
            ROW=ROW+2
            DO 206 I = 1,NUM
               F1(I)=F(C,I+LLIM-1,4,F1EST(I))
206         CONTINUE
            CALL LINER(PAGE2(ROW),FM(2),1.0,0,TCP,F1,F1EST,NUM,1)
            IF (RNUM.EQ.4) THEN
               ROW=ROW+2
               DO 221 I = 1,NUM
                  F1(I)=F(C,I+LLIM-1,1,F1EST(I))
221            CONTINUE
               CALL LINER(PAGE2(ROW),FM(2),1.0,0,TCW,F1,F1EST,NUM,1)
            ELSEIF (RNUM.EQ.5) THEN
               ROW=ROW+1
               DO 207 I = 1,NUM,1
                  F1(I) = GETLEN(I+LLIM-1)
                  F1EST(I)=' '
207            CONTINUE
               CALL LINER(PAGE2(ROW),FM(1),TOL,0,DIS, F1,F1EST,NUM,2)

               ROW = ROW + 2
               DO 208 I = 1,NUM,1
                  F1(I)=F(C,I+LLIM-1,RNUM,F1EST(I))
208            CONTINUE
               CALL LINER(PAGE2(ROW),FM(1),TOL,0,TC1,F1,F1EST,NUM,2)
            ELSEIF (RNUM.EQ.6) THEN
               ROW=ROW+1
               DO 209 I = 1,NUM,1
                  F1(I)=GETDUR(I+LLIM-1)*GETLEN(I+LLIM-1)
                  F1EST(I) = ' '
209            CONTINUE
               CALL LINER(PAGE2(ROW),FM(1),TOL,0,DISH,F1,F1EST,NUM,2)
               ROW=ROW+2
               DO 210 I = 1,NUM,1
                  F1(I)=F(C,I+LLIM-1,RNUM,F1EST(I))
210            CONTINUE
               CALL LINER(PAGE2(ROW),FM(1),TOL,0,TC2,F1,F1EST,NUM,2)
            ENDIF
         ENDIF
      RETURN
      END
```